KALI LINUX
ОТ РАЗРАБОТЧИКОВ

Рафаэль Херцог, Джим О'Горман, Мати Ахарони



Kali Linux Revealed

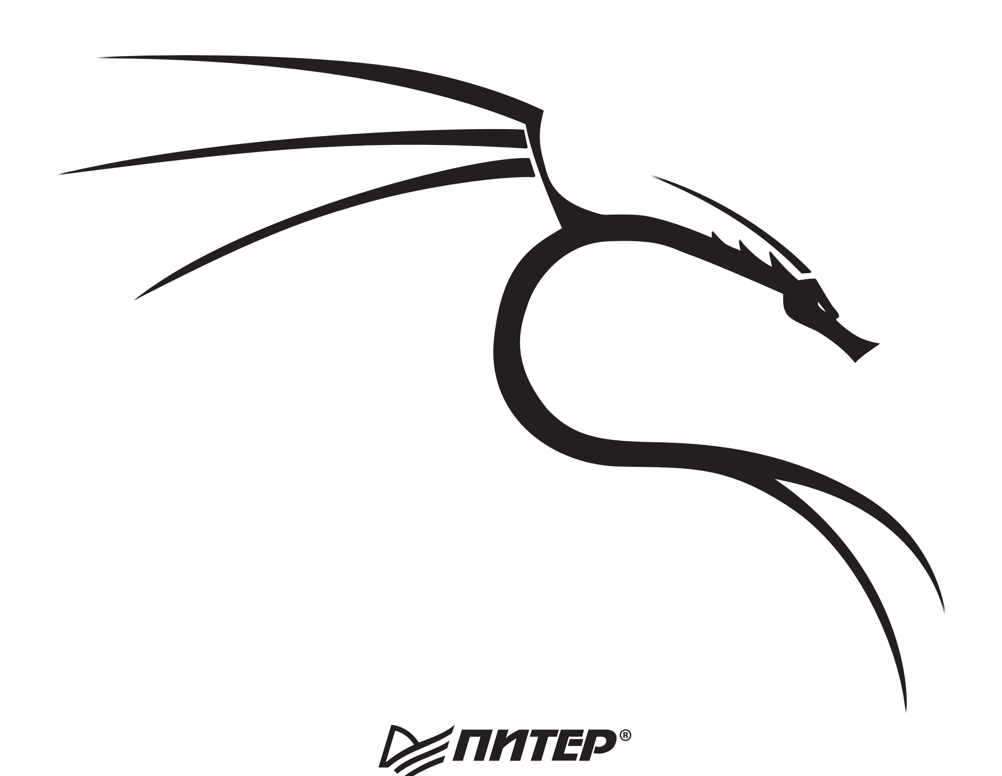
Mastering the Penetration Testing Distribution

by Raphaël Hertzog, Jim O'Gorman, and Mati Aharoni



Рафаэль Херцог, Джим О’Горман, Мати Ахарони

KALI LINUX
ОТ РАЗРАБОТЧИКОВ



ПИТЕР®

Санкт-Петербург • Москва • Екатеринбург • Воронеж
Нижний Новгород • Ростов-на-Дону
Самара • Минск

2019

ББК 32.973.2-018.2
УДК 004.451
Х39

Херцог Рафаэль, О'Горман Джим, Ахарони Мати

Х39 Kali Linux от разработчиков. — СПб.: Питер, 2019. — 320 с.: ил. — (Серия «Для профессионалов»).

ISBN 978-5-4461-0548-9

Авторы шаг за шагом познакомят вас с основами и возможностями Kali Linux.

В книге предложен краткий курс работы с командной строкой Linux и ее концепциями, описаны типичные сценарии установки Kali Linux. Прочитав эту книгу, вы научитесь конфигурировать, отлаживать и защищать Kali Linux, а также работать с мощным менеджером пакетов дистрибутива Debian. Научитесь правильно устанавливать Kali Linux в любых окружениях, в том числе в крупных корпоративных сетях. Наконец, вам предстоит познакомиться и со сложными темами: компиляцией ядра, созданием собственных образов ISO, промышленным шифрованием и профессиональной защитой конфиденциальной информации.

16+ (В соответствии с Федеральным законом от 29 декабря 2010 г. № 436-ФЗ.)

ББК 32.973.2-018.2
УДК 004.451

ISBN 978-0997615609 англ.
ISBN 978-5-4461-0548-9

Краткое содержание

Оглавление

Предисловие

Вы не представляете, насколько здорово, что вы сейчас читаете эту книгу.

В 1998 году я был подающим надежды хакером, а также одним из сооснователей профессиональной команды «белых шляп» (хакеров, не нарушающих закон). Мы были детьми, получившими работу мечты. Нам платили за проникновения в самые защищенные компьютерные системы, сети и здания мира.

Звучит весьма заманчиво, но на самом деле большую часть времени мы проводили, нависая над клавиатурой, вооружившись цифровыми инструментами нашего ремесла. В нашем распоряжении была убогая коллекция программ, созданных, чтобы находить сети и обнаруживать цели, затем сканировать их, эксплуатировать и менять их задачи в наших интересах. В некоторых случаях один из нас (зачастую Джим Чаппел) мог написать специальную утилиту для грязных делишек типа сканирования сетей класса A (то, чего в то время не могло сделать ни одно другое средство), но чаще мы использовали или модифицировали инструменты, созданные хакерским сообществом. В те дни, когда еще не существовало Google, мы часто посещали BugTraq, AstaLaVista, Packet Storm, w00w00, SecurityFocus, X-Force и прочие ресурсы для проведения исследований и построения собственного арсенала.

Поскольку на каждое задание отводилось ограниченное время, нам приходилось работать быстро. Это значило, что мы не могли долго возиться с утилитами. А значит, мы вынуждены были изучить основные инструменты вдоль и поперек, а вспомогательные на всякий случай держать под рукой. И наш инструментарий нам следовало содержать в полном порядке, документированным и протестированным, чтобы во время работы иметь как можно меньше сюрпризов. В конечном счете, если проникнуть не удавалось, мы теряли лицо перед клиентами и наши рекомендации воспринимались менее серьезно.

Из-за этого я тратил немало времени на каталогизацию инструментов. Таким образом, как только выходили новые программы или появлялись обновления для уже доступных, начиналась моя рутинная деятельность. Я обязан был выяснить, будет ли инструмент функционировать на платформе атаки (часть из них не работала) и заслуживает ли он внимания вообще (некоторые того не стоили).

Мне приходилось обновлять каждый сценарий, зависящий от этого инструмента, документировать его и тестировать, включая перенос любых изменений, внесенных в предыдущих версиях.

Затем я реорганизовывал все инструменты и помещал их в различные каталоги согласно их предназначению. Я должен был писать сценарии-оболочки для определенных инструментов, объединять часть из них и помещать все это на отдельный компакт-диск, который мы могли бы использовать в уязвимых областях, когда клиенты не позволяли нам применять машины атаки или извлекать носители данных из своих лабораторий.

Этот процесс был многострадальным, но необходимым. Мы знали, что у нас имелась возможность проникнуть в любую сеть: достаточно было только надлежащим образом применить наши навыки и опыт, оставаться организованными и работать эффективно. И хотя желание оставаться непобежденными нас очень мотивировало, все же речь шла о предоставлении услуг клиентам, которые *нуждались* в них для проникновения в различные сети таким образом, чтобы они могли заполнить пробелы и перевести деньги на критически важные, но должным образом неоцененные программы, связанные с информационной безопасностью.

На то, чтобы отточить наши навыки и приобрести бесценный опыт, были потрачены годы, но мы не добились бы этого успеха без должной организации и эффективности. Мы однозначно потерпели бы неудачу, не окажись необходимые инструменты в нужный момент под рукой.

Именно поэтому я тратил много времени на проведение различного рода исследований, проверку и каталогизацию инструментов, и на пороге XXI века все перечисленное очень быстро стало непростой работой с полной занятостью. С распространением Интернета разнообразие атак по всему миру значительно увеличилось и, соответственно, количество инструментов для проведения атак возросло экспоненциально, как и количество усилий, необходимых для их поддержания.

Начиная с 2004 года Интернет стал стремительно развиваться не только как база для многих вариантов ведения бизнеса, но и как прогрессивная социальная платформа. Компьютеры стали доступными, повсеместными и простыми в использовании. Технология хранения была расширена от мегабайтов до гигабайтов. Ethernet вырос с сотен килобит до десятков мегабит в секунду, а интернет-соединения стали быстрее и дешевле, чем когда-либо прежде. Электронная коммерция была на подъеме, появились такие социальные сети, как Facebook (2004) и Twitter (2006), а Google (1998) возмужал настолько, что любой (в том числе и преступник) мог найти в Интернете что угодно.

Проведение исследований стало критически необходимым действием для команд, подобных нашей, поскольку нам приходилось идти в ногу с новыми видами атак и наборами инструментов. Мы имели дело с большим количеством компьютерных преступлений, и криминалистическая работа требовала, чтобы мы действовали осторожно, не повредив потенциальные улики. Концепция *live CD* позволяла нам

провести полноценную экспертизу на подвергшемся атаке компьютере без ущерба для улик.

Теперь нашей маленькой команде приходилось управляться с инструментами атаки, криминалистическими утилитами и распределением программ уязвимых зон. Мы должны были не отставать от всех последних методов проведения атак и взломов; и нам приходилось делать то, за что нам, собственно говоря, и платили, — проводить тесты на проникновение, которые пользовались большим спросом. Все выходило из-под контроля, и вскоре мы проводили уже меньше времени непосредственно в бою и гораздо больше — за исследованиями, оттачиванием наших инструментов и детальным планированием.

Мы были не одни в этой борьбе. В 2004 году Мати Мутс Ахарони, хакер и специалист по безопасности, выпустил WHoppiX (White Hat Knoppix), live CD для Linux, который назвал «Окончательная версия live CD для тестирования на проникновение». Этот инструмент содержал «все эксплойты из SecurityFocus, Packet Storm и k-otik, Metasploit Framework 2.2 и др.».

Я помню, как скачивал WHoppiX и думал, насколько это замечательная вещь. Я скачал другие образы live CD и думал, что если когда-либо попаду в некую передрягу, эти диски смогут реально спасти мою шкуру. Тем не менее я не особо рассчитывал на WHoppiX или любые другие диски в реальной работе. Я не доверял ни одному из них, выполняя свои основные задачи; ни один из них не подходил для моего рабочего процесса; все они не были полными устанавливаемыми дистрибутивами и в момент их скачивания уже являлись устаревшими. Использование устаревших наборов инструментов в нашей сфере деятельности равносильно поцелую смерти.

Я просто добавил эти образы компакт-дисков, несмотря на их относительно большие размеры, в наш арсенал и продолжил болезненный процесс поддержания нашего «реального» инструментария.

Однако несмотря на мое личное мнение в тот момент и, возможно, несмотря на ожидания Мутса, WHoppiX и его «потомки» произвели сейсмический толчок в жизни Мутса, нашей сфере деятельности и нашем сообществе.

В 2005-м WHoppiX эволюционировал в WHAX с расширенным и обновленным набором инструментов, основанным на «более модульном live CD SLAX (Slackware)». Мутс и растущая команда волонтеров из сообщества хакеров, казалось, понимали: независимо от их уровня проницательности, они никогда не могли предвидеть весь рост и постоянную изменчивость нашей сферы деятельности и то, что пользователи их компакт-дисков будут иметь абсолютно разные потребности в этой области. Представлялось очевидным, что Мутс и его команда фактически использовали WHAX в реальной работе и были заинтересованы в том, чтобы он функционировал должным образом. Это меня очень воодушевляло.

В мае 2006 года Мутс и Макс Мозер объединили Auditor Security Linux и WHAX в один дистрибутив под названием BackTrack. Все еще основываясь на SLAX, BackTrack продолжал расти, включая в себя больше утилит и фреймворков, рас-

ширенную языковую поддержку, широкую беспроводную поддержку, структуру меню, обслуживающую как начинающих, так и профессиональных пользователей, и сильно модифицированное ядро. BackTrack стал ведущим дистрибутивом в области безопасности, но многие, как я, все еще применяли его в качестве резервной копии для своих «реальных инструментов».

К началу 2009 года Мутс и его команда значительно расширили BackTrack до версии 4. Будучи постоянной работой для Мутса, BackTrack больше не был просто live CD, а являл собой полноценный дистрибутив Ubuntu, использующий репозитории программного обеспечения (ПО) Ubuntu. Данное изменение означало серьезную эволюцию: BackTrack 4 имел механизм обновления. По словам самого Мутса: «При синхронизации с репозиториями BackTrack вы будете регулярно получать обновления средств безопасности вскоре после их выпуска».

Это был переломный момент. Команда BackTrack настроилась на ту борьбу, которую вели все специалисты по тестированию на проникновение, криминалистические аналитики и прочие специалисты, работающие в нашей сфере. Их усилия сэкономили нам много времени и предоставили твердую и уверенную основу, позволяя нам возвращаться в бой и проводить больше времени над решением важных (и забавных) задач. В результате сообщество отреагировало активной дискуссией на форумах и в «Вики»; многие подключились к команде разработчиков. BackTrack был поистине общественным детищем с Мутсом во главе.

BackTrack 4 наконец стал промышленно мощной платформой, и мы с коллегами смогли вздохнуть с облегчением. Мы прекрасно осознавали всю «боль и страдание», которое испытывал Мутс и его команда, поскольку все мы были на их месте. В результате многие из нас начали использовать BackTrack в качестве базовой основы для нашей деятельности. Да, мы по-прежнему работали с инструментами, писали собственный код и разрабатывали личные приемы и различные техники; мы продолжали исследовать и экспериментировать, но не тратили все время на сбор, обновление, проверку и организацию утилит.

В 2010 году BackTrack 4 R1 и R2 стали очередными изменениями, приведшими к восстановлению с нуля BackTrack 5 в 2011 году. Продолжая основываться на Ubuntu и набирая обороты с каждым новым выпуском, BackTrack стал массивным проектом, который теперь уже требовал не только огромных усилий со стороны волонтеров и сообщества, но и серьезного финансирования. Мутс запустил Offensive Security (в 2006 году) не только для предоставления тренинга мирового класса и услуг тестирования на проникновение, но и для того, чтобы обеспечить механизм разработки BackTrack и убедиться в том, что BackTrack по-прежнему останется с открытым исходным кодом и бесплатным в использовании.

BackTrack продолжал развиваться и улучшаться до 2012 года (с R1, R2 и R3), поддерживая ядро Ubuntu и пополняясь сотнями новых инструментов, включая средства физического и аппаратного обеспечения, поддержку VMware, бесчисленные беспроводные и аппаратные драйверы и множество улучшений стабильности и исправления ошибок. Однако после выпуска R3 разработка BackTrack продолжила развиваться как-то подозрительно спокойно.

Сразу же появилось несколько основных предположений относительно того, что же происходит на самом деле. Некоторые считали, будто BackTrack за огромную сумму «продал душу» некоему злобному корпоративному повелителю. Со временем Offensive Security превратилась в одну из самых уважаемых обучающих компаний, а также стала лидером новаторских мыслей в нашей сфере, и кое-кто судачил, что подобный успех поглотил и оттеснил на второй план ведущих разработчиков BackTrack. Однако ничто не могло быть настолько далеким от правды.

В 2013 году была выпущена версия Kali Linux 1.0. Из примечаний к выпуску: «Через год безмолвного развития Offensive Security с гордостью объявляет о выпуске и общей доступности в сети Kali Linux — самого передового, надежного и стабильного дистрибутива для проведения тестов на проникновение на сегодняшний день. Kali — более зрелая, безопасная и готовая к работе версия BackTrack».

Kali Linux не был простым ребрендингом BackTrack. Включая в себя более 600 полностью переупакованных инструментов, он являл собой просто потрясающий набор утилит, но это еще не все. Kali был создан с нуля на основе Debian. Для неосведомленных данный факт может показаться не очень большим достижением. Благодаря огромным усилиям по переупаковке инструментов пользователи Kali могли загрузить исходный код для каждого отдельного средства; они могли модифицировать и перестраивать его по мере необходимости всего лишь несколькими нажатиями клавиш. В отличие от других основных современных операционных систем, Kali Linux синхронизировался с репозиториями Debian четыре раза в день, а это значило следующее: пользователи могли получить актуальные обновления пакетов и необходимые исправления безопасности. Разработчики прилагали огромные усилия для создания новых пакетов и поддержки более свежих версий инструментов таким образом, чтобы пользователи всегда имели доступ к любым обновлениям. Благодаря Debian-корням Kali пользователи могли загружать дистрибутив или ISO-образ непосредственно из репозиториев, что предоставило новые возможности для полностью настроенных Kali-установок или массивных развертываний предприятий, которые можно было бы дополнительно автоматизировать и настроить с помощью файлов-пресетов. И в завершение настраиваемого трио пользователи Kali могли изменять среду рабочего стола, меню, значки и даже заменять среды окон. Широкое развитие ARM создало предпосылки для установки Kali Linux на большой спектр аппаратных платформ, включая точки доступа, одноплатные компьютеры (например, Raspberry Pi, ODROID, BeagleBone и CubieBoard) и компьютеры Chromebook на базе ARM. И последнее, но не по важности: Kali Linux поддерживает плавные маленькие и большие обновления, а это значит, что приверженцам Kali никогда не придется переустанавливать уже настроенные пользовательские установки Kali Linux.

Сообщество обратило внимание, что в первые пять дней 90 000 человек загрузили Kali 1.0.

Это было только начало. В 2015 году был выпущен Kali 2.0, за которым следуют календарные релизы в 2016 году. Вкратце: «Если Kali 1.0 был ориентирован на создание прочной инфраструктуры, то Kali 2.0 в основном сосредоточен на том,

чтобы реорганизовывать опыт пользователя и поддерживать обновленные пакеты и репозитории инструментов».

Современная версия Kali Linux является динамическим дистрибутивом, что само по себе знаменует конец дискретных версий. Теперь пользователи постоянно обновляются и получают обновления и исправления по мере появления последних. Основные инструменты обновляются чаще благодаря более новой версии системы ярлыков, также были реализованы революционные улучшения доступности для слабовидящих или полностью слепых людей, а ядра Linux обновлены и исправлены в целях продолжения поддержки беспроводной инъекции 802.11. Software Deffined Radio (SDR) и Near-Field Communication (NFC) добавляет поддержку для новых областей тестирования безопасности. Благодаря Linux LV и LUKS доступны дополнительные возможности установки зашифрованных дисков Linux, а также добавлены опции USB-устойчивости, позволяющие поддерживать изменения между перезагрузками, для установки Kali с USB-накопителя независимо от того, зашифрован он или нет. И наконец, последние версии Kali открыли дверь для NetHunter — операционной системы мирового класса с открытым исходным кодом, работающей на мобильных устройствах на базе Kali Linux и Android.

Дистрибутив Kali Linux эволюционировал не просто в платформу для профессионалов информационной безопасности, а в полноценный, безопасный, оперативный и созревший дистрибутив мирового класса.

На протяжении десяти лет Мутс и его команда вместе с бесчисленным множеством волонтеров из хакерского сообщества занимались оптимизацией и организацией нашей рабочей среды, таким образом освободив нас от монотонных операций, предоставив безопасный и надежный фундамент, позволив сконцентрироваться на продвижении нашей сферы деятельности к финальной цели — обеспечению безопасности нашего цифрового мира.

Очень интересно, но неудивительно, что вокруг Kali Linux образовалось потрясающее сообщество. Каждый месяц от 300 до 400 тысяч новых пользователей скачивают версию Kali. Мы собираемся на Kali-форумах сорокатысячным сообществом, а 300–400 из нас можно одновременно найти на IRC-канале Kali. Мы встречаемся на конференциях и посещаем Kali Dojos, чтобы узнать у самих разработчиков, как лучше применять Kali.

Kali Linux изменил мир информационной безопасности к лучшему, и Мутс со своей командой спасли каждого из нас от бесчисленных часов работы и разочарования, что позволило нам тратить больше времени и энергии на совместное развитие нашей сферы деятельности.

Однако несмотря на удивительное признание, поддержку и популярность, к Kali никогда не прилагалось официальное руководство. Теперь ситуация изменилась. Я очень рад, что вместе с командой разработчиков Kali (Мати Ахарони, Рафаэлем Херцогом, Девоном Кирнсом и Джимом О'Горманом) могу предложить вам это руководство — возможно, первое издание в предстоящей серии публикаций, посвященных Kali Linux. В данной книге мы сосредоточимся на самом дистрибутиве

Kali Linux, который поможет вам понять и максимально упростить использование Kali с нуля. Тем не менее пока мы не будем углубляться в арсенал инструментов, содержащихся в Kali Linux. Однако, независимо от того, являетесь ли вы ветераном или абсолютным новичком в данной сфере, начинать лучше отсюда, если вы готовы разбираться в Kali Linux и относиться к нему со всей серьезностью. Независимо от того, сколько времени вы работаете в этой сфере, ваше решение прочитать книгу связывает вас с растущим сообществом Kali Linux — одним из старейших, крупнейших, активнейших и ярчайших в нашей отрасли.

От имени Мутса и остальной части потрясающей команды Kali поздравляю вас с первым шагом к освоению Kali Linux!

Джонни Лонг,
февраль 2017

Вступление

Шестнадцать профессиональных ноутбуков, заказанных для вашей команды специалистов по тестированию на проникновение, только что прибыли, и вам поставлена задача настроить их для дистанционного использования. Вы установили Kali и запустили один из ноутбуков, чтобы выяснить, работает ли он. Несмотря на новейшее ядро Kali, сетевая карта и мышь не функционируют, а мощная видеокарта NVIDIA и GPU выводят изображение ужасного качества, так как для них не установлены драйверы. Вы вздыхаете.

В Kali в *режиме реального времени* вы быстро набираете в оболочке командной строки `lspci` и ждете. Прокручиваете список аппаратного обеспечения: «PCI-мост, USB-контроллер, SATA-контроллер. Ага! Контроллеры сети и Ethernet». Быстрый поиск в Google соответствующих номеров моделей, соотнесенных с версией Kali, — и оказывается, что эти драйверы еще не попали в основное ядро.

Но не все потеряно. В вашей голове постепенно созревает план, и вы благодарите небеса за эту книгу, которую приобрели пару недель назад. Вы можете применить live-сборку системы Kali, чтобы создать пользовательский ISO-образ Kali, в который будут входить необходимые драйверы, встроенные в систему установки. Кроме того, можете добавить драйверы для видеокарты NVIDIA, а также библиотеки CUDA, нужные для того, чтобы этот GPU «мило общался» с хешкэтом (hashcat) и не мешал быстро взламывать пароли. Хо-хо, вы можете даже внедрить пользовательские обои с логотипом Microsoft, чтобы подразнить свою команду.

Так как технические профили для вашей установки идентичны, вы добавляете опцию автоматического запуска для ISO, чтобы ваша команда могла подключить USB и установить Kali без вмешательства пользователя — установка сама позаботится о себе, о шифровании дисков и т. д.

Великолепно! Теперь вы можете создавать по первому требованию усовершенствованную версию Kali, спроектированную и оптимизированную под ваше аппаратное обеспечение. Вы сэкономили целый день. Миссия выполнена!

Учитывая огромное количество аппаратного обеспечения, наводнившего рынок, данный сценарий становится более приемлемым для тех, кто хочет уйти от основных

операционных систем в поиске чего-то более экономного, среднего или более соответствующего нашей работе и стилю.

Вышесказанное особенно применимо к тем, кого привлекает область безопасности, будь то хобби, увлечение или постоянная работа. Новички часто заходят в тупик из-за окружения или операционной системы. Для многих новичков Kali — первый шаг к Linux.

Мы заметили эту тенденцию в нашей базе пользователей пару лет назад и поняли, что можем помочь им, создав структурированную книгу, которая упростит ориентацию в мире безопасности, предоставляя знания о Linux, необходимые для начала работы.

В итоге мы остались довольны результатами. Книга соответствует всем нашим требованиям, и я с гордостью могу сказать, что она превзошла ожидания. Мы поняли, что увеличили базу пользовательского потенциала книги. Она больше не предназначается для новичков в области безопасности и включает много полезной информации для опытных специалистов, которым необходимо улучшить их контроль над Kali Linux, что позволит им раскрыть весь потенциал нашего продукта. Будут ли они тестировать одну машину или тысячи по всему предприятию, вносить незначительные изменения либо полностью модифицировать программное обеспечение под заказчика, создавать собственные репозитории или углубляться в удивительный пакет системы управления Debian, — эта книга всегда предоставит план действий.

С вашим путеводителем в руках, от своего имени и имени всей команды Kali Linux я желаю вам захватывающего, веселого, плодотворного и «раскрывающего» путешествия!

Мутс, февраль 2017

Введение

Kali Linux — самый мощный и популярный в мире дистрибутив для тестирования на проникновение, используемый профессионалами безопасности в широком спектре специальностей, включая тестирование на проникновение, криминалистику, обратное проектирование и оценку уязвимости. Это кульминация многолетних усовершенствований и результат непрерывной эволюции от WHoppiX к WHAX, затем к BackTrack, а теперь к полноценному дистрибутиву тестирования на проникновение. В Kali применяются многие функции Debian GNU/Linux и учитываются ценные советы членов динамичного мирового сообщества, работающего над ПО с открытым исходным кодом.

Kali Linux был создан не как простая коллекция инструментов, а скорее как гибкая структура, в которой специалисты по тестированию на проникновение, энтузиасты в сфере безопасности, студенты и любители могут настроить утилиты в соответствии с конкретными потребностями.

Почему именно эта книга?

Kali Linux — это не просто набор различных средств информационной безопасности, которые установлены на стандартной платформе Debian и предварительно настроены, чтобы позволить незамедлительно начать работу. Для получения максимальной отдачи от Kali важно иметь полное представление о возможностях его оснований Debian GNU/Linux (которые поддерживают все эти прекрасные инструменты) и изучить, каким образом вы можете использовать их в своей среде.

Хотя Kali определенно многоцелевой дистрибутив, он в первую очередь предназначен для оказания помощи в тестировании на проникновение. Цель этой книги состоит в том, чтобы не только помочь вам чувствовать себя увереннее в момент использования Kali Linux, но также углубить ваше понимание и оптимизировать опыт. При проведении теста на проникновение и ограниченном времени вам не придется переживать о потере драгоценных минут, уходящих на установку нового программного обеспечения или включение нового сетевого сервиса. В этой книге мы

познакомим сначала с Linux, а затем, окунувшись чуть глубже, рассмотрим нюансы, характерные для Kali Linux, чтобы вы точно понимали всю суть происходящего.

Это бесценные знания, особенно если вам приходится работать в условиях ограниченного времени. Они обязательно пригодятся, когда вы будете настраивать программу, устранять проблему, пытаться сконфигурировать инструмент по своему усмотрению, анализировать вывод утилиты или использовать Kali в расширенной среде.

Для вас ли эта книга?

Если вы хотите погрузиться в интеллектуально богатую и невероятно увлекательную область информационной безопасности и по праву выбрали Kali Linux в качестве основного дистрибутива, то эта книга поддержит вас в данном путешествии. Она была написана, чтобы помочь тем, кто использует Linux впервые, а также пользователям Kali, стремящимся углубить свои знания об основах Kali, и тем, кто уже много лет применяет Kali, но хочет формализовать свое обучение, расширить знания об этом дистрибутиве и заполнить пробелы в знаниях.

Кроме того, эта книга может служить путеводителем, технической ссылкой и учебным пособием для тех, кто готовится к сертификации Kali Linux Certified Professional.

Общий подход и структура издания

Книга структурирована таким образом, что вы можете приступить к использованию Kali Linux с самого начала ее прочтения. Вам не придется читать до середины книги, прежде чем начать практиковаться. Каждая тема рассматривается очень подробно, и издание наполнено примерами и снимками экрана, призванными сделать объяснения более наглядными.

В главе 1 «О Kali Linux» мы определим некую базовую терминологию и объясним предназначение Kali Linux. В главе 2 «Начало работы с Kali Linux» мы шаг за шагом проведем вас от загрузки ISO-образа к запуску этого дистрибутива на вашем компьютере. Далее следует глава 3 «Основы Linux», в которой предоставлены необходимые базовые знания о системе Linux, такие как ее архитектура, процесс установки, иерархия файловой системы, разрешения и т. д.

После прочтения первых трех глав вы будете уметь использовать Kali Linux лишь в качестве live-системы. В главе 4 «Установка Kali Linux» вы узнаете, как выполнить постоянную установку дистрибутива (на ваш жесткий диск), и в главе 5 «Настройка Kali Linux» — как настроить его по вашему усмотрению. Став полноценным пользователем Kali, вы получили возможность ознакомиться с важными ресурсами: глава 6 «Самостоятельное решение проблем и получение помощи» даст вам ключи для решения непредвиденных проблем, с которыми, вероятно, придется столкнуться.

Поскольку основы освещены достаточно подробно, в остальной части книги объясняются более продвинутые темы: глава 7 «Защита и контроль Kali Linux»

поможет удостовериться, что установка дистрибутива соответствует вашим требованиям безопасности. Далее, в главе 8 «Управление пакетами Debian» объясняется, как применять весь потенциал системы пакетов Debian. И в главе 9 «Расширенное использование системы» вы узнаете, как создать полностью настроенный ISO-образ Kali Linux. Все эти темы еще более актуальны, когда вы переходите к полноценной работе с дистрибутивом, как описано в главе 10 «Kali Linux в организации».

Глава 11 «Оценка защищенности информационных систем» устанавливает связь между всем, что вы узнали в этой книге, и повседневной работой специалистов по безопасности.

Благодарности от Рафаэля Херцога

Я хотел бы поблагодарить Мати Ахарони: в 2012 году он связался со мной, поскольку я был одним из десятков консультантов Debian, а ему нужно было создать преемника BackTrack, который основывался бы на Debian. Так я начал работать над Kali Linux, и мне понравилось путешествие в мир Kali.

За прошедшие годы Kali Linux приблизился к Debian GNU/Linux, особенно с переходом к Kali Rolling, основанному на Debian Testing. Теперь большая часть моей работы, на Kali или Debian, обеспечивает преимущества для всей системы Debian. И это именно то, что мотивирует меня продолжать, изо дня в день, месяц за месяцем, из года в год.

Работа над данной книгой также является прекрасной возможностью, которую предложил Мати. Это совершенно другой вид деятельности, но все такой же полезный: помогать людям и делиться с ними своими знаниями об операционной системе Debian/Kali. Основываясь на моем опыте с книгой Debian Administrator's Handbook, я надеюсь, что мои разъяснения помогут вам начать работу в быстро развивающемся мире компьютерной безопасности.

Я также хотел бы поблагодарить всех представителей Offensive Security, которые были вовлечены в работу над этой книгой: Джима О'Гормана (соавтора отдельных глав), Девона Кирнса (рецензента), Рона Генри (технического редактора), Джо Штейнбаха и Тони Круза (руководителей проекта). И спасибо Джонни Лонгу, который присоединился, чтобы написать предисловие, но в итоге работал над всей книгой.

Благодарности от Джима О'Гормана

Я хотел бы поблагодарить всех, кто участвовал в этом проекте, за их вклад, в котором лишь малая часть моя. Данная книга так же, как и сам дистрибутив Kali Linux, стала совместным проектом множества людей, выполняющих простую работу. Особая благодарность Рафаэлю, Девону, Мати, Джонни и Рону за то, что взяли на себя львиную долю усилий. Без них книга не появилась бы.

Благодарности от Мати Ахарони

Прошло несколько лет с тех пор, как Kali Linux был впервые выпущен, и с первого дня я всегда мечтал опубликовать официальную книгу, которая охватывает всю операционную систему Kali в целом. Поэтому для меня большая честь наконец видеть книгу, которая сделает Kali доступным для общественности. Я хотел бы искренне поблагодарить всех, кто участвует в создании данного проекта, включая Джима, Девона, Джонни и Рона. Особая благодарность Рафаэлю за то, что он сделал большую часть тяжелой работы в этой книге и привнес свой огромный опыт в нашу группу.

ГЛАВА

1

О Kali Linux



Ключевые темы:

- дистрибуция Linux;
- производный дистрибутив Debian;
- предназначение;
- характеристики;
- политики.

Kali Linux (https://www.kali.org/) — это дистрибутив Linux для проверки корпоративной безопасности, основанный на Debian GNU/Linux. Операционная система Kali предназначена для специалистов по безопасности и ИТ-администраторов и позволяет им проводить профессиональное тестирование на проникновение в систему, информационный криминалистический анализ и аудит безопасности информационных систем.

Что такое дистрибутив Linux

Хотя обычно так называют целую операционную систему, Linux — это лишь название ядра, части программного обеспечения, которая обрабатывает взаимодействия между оборудованием и конечными пользовательскими приложениями.

В то же время выражение «дистрибутив Linux» определяет в целом операционную систему, построенную на ядре Linux, обычно содержащую программу установки и множество приложений, которые либо предварительно установлены, либо пакетированы для легкой установки.

Debian GNU/Linux (https://www.debian.org/) — ведущий универсальный дистрибутив Linux, известный своим качеством и стабильностью. Kali Linux основывается на работе проекта Debian, добавляя более 300 собственных специализированных пакетов, связанных с информационной безопасностью, особенно в области тестирования на проникновение.

Debian — это проект, относящийся к свободно распространяемому ПО и предоставляющий несколько версий операционной системы. Для обозначения его конкретной версии часто используют термин «дистрибутив», скажем, дистрибутивы Debian Stable или Debian Testing. То же самое относится и к Kali Linux — например, дистрибутив Kali Rolling.

1.1. Немного истории

Проект Kali Linux плавно стартовал в 2012 году, когда специалисты Offensive Security решили заменить свой почтенный проект Linux BackTrack, который поддерживался вручную, чем-то, что могло бы стать настоящим дериватом Debian (https://wiki.debian.org/Derivatives/Census), дополненным всеми необходимыми объектами и улучшенными методами пакетирования. Было принято решение построить Kali поверх дистрибутива Debian, известного своим качеством, стабильностью и широким выбором доступного программного обеспечения. Вот почему я (Рафаэль) участвовал в этом проекте как консультант Debian.

Первый релиз (версия 1.0) вышел год спустя, в марте 2013 года, и был основан на Debian 7 Wheezy, стабильном (в то время) дистрибутиве Debian. В течение первого года разработки мы компоновали сотни приложений, связанных с тестированием на возможность проникновения, и создавали инфраструктуру. И хотя количество приложений имеет значение, они были тщательно отобраны — мы отбросили те

из них, которые больше не работали, а также те, что дублировали функции, уже доступные в более эффективных инструментах.

На протяжении двух лет после выхода версии 1.0 у Kali появилось множество дополнительных обновлений, которые расширили диапазон доступных приложений и улучшили аппаратную поддержку благодаря новым версиям ядра. Вкладывая силы и средства в непрерывную интеграцию, мы стремились гарантировать, что все важные пакеты готовы к установке и пользователь всегда сможет создать собственные live-образы (отличительная черта дистрибутива).

В 2015 году, когда вышел релиз Debian 8 Jessie, шла работа над переносом на его базу Kali Linux. Хотя система Kali Linux 1.x обходилась без GNOME (используя вместо этого GNOME Fallback), в новой версии мы решили охватить и улучшить эту оболочку. В частности, мы добавили некоторые расширения GNOME для внедрения недостающих функций, в первую очередь меню Applications (Приложения). Результатом нашей деятельности стал дистрибутив Kali Linux 2.0, выпущенный в августе 2015 года.

GNOME — рабочая среда Kali Linux по умолчанию

Рабочая среда (среда рабочего стола, настольная среда) представляет собой набор графических приложений, которые используют общий графический инструментарий и предназначаются для совместного применения на рабочих местах пользователей. На серверах такие среды обычно отсутствуют. Чаще всего они предоставляют программу запуска приложений, менеджер файлов, браузер, почтовый клиент, офисный пакет и т. д.

GNOME (https://www.gnome.org/) — одна из самых популярных сред рабочего стола (вместе с KDE (https://www.kde.org/), Xfce (https://xfce.org/), LXDE (https://lxde.org/), MATE (http://mate-desktop.org/)), устанавливается на основные ISO-образы, предоставляемые Kali Linux. Если вам не нравится GNOME, то можете легко создать пользовательский ISO-образ со средой рабочего стола по вашему выбору. Инструкции описаны в главе 9.

В то же время мы еще тщательнее проработали инструменты, отвечающие за защиту от несанкционированного доступа, с целью гарантировать, что Kali Linux всегда содержит последнюю версию приложений для тестирования на проникновение. К сожалению, данная задача немного расходилась с использованием Debian Stable в качестве основы дистрибутива, поскольку для ее выполнения требовалось резервировать множество пакетов. Это связано с тем, что в Debian Stable приоритетом является стабильность программного обеспечения, результатом чего выступает большой промежуток от момента выпуска обновления до его интеграции в дистрибутив. Учитывая наше стремление к непрерывной интеграции, было вполне естественным шагом перенести Kali Linux на базу Debian Testing, чтобы мы могли воспользоваться последней версией пакетов Debian, как только они становились доступными. У Debian Testing гораздо более интенсивный цикл обновления, который оптимально соответствует философии Kali Linux.

По сути, это концепция релиза Kali Rolling. В то время как дистрибутивы с плавающими релизами были доступны в течение уже довольно длительного времени, Kali 2016.1 стал первым релизом, официально принявшим плавающий характер. Когда вы устанавливаете последнюю версию Kali, ваша система фактически отслеживает дистрибутив Kali Rolling и *каждый день вы получаете новые обновления*. Раньше выпуски Kali представляли собой снимки базового дистрибутива Debian с установленными в него пакетами Kali.

Дистрибутив с плавающим релизом имеет много преимуществ, но сопряжен и со множеством проблем как для тех из нас, кто работает над ним, так и для пользователей, которым приходится справляться с бесконечным потоком обновлений, а иногда и обратно-несовместимыми изменениями. Эта книга предоставит вам информацию, необходимую для решения всех проблем, которые могут возникнуть при управлении установкой Kali Linux.

1.2. Взаимоотношения с Debian

Дистрибутив Kali Linux основан на версии Debian Testing (https://www.debian.org/releases/testing/) (это текущее состояние разработки следующего стабильного дистрибутива Debian). Как следствие, большинство пакетов, доступных в Kali Linux, исходят прямо из репозитория Debian.

Хотя Kali Linux в значительной степени зависит от Debian, он также полностью независим в том смысле, что есть собственная инфраструктура, в которую можно вносить любые изменения по своему желанию.

Движение пакетов

Создатели Debian ежедневно ведут работу над обновлением пакетов и загрузкой их в дистрибутив Debian Unstable. Оттуда пакеты переносятся в дистрибутив Debian Testing сразу после удаления самых вредоносных ошибок. Процесс переноса также гарантирует нерушимость каких-либо зависимостей в Debian Testing. Задача состоит в том, чтобы поддерживать Debian Testing всегда в удобном для пользователя (или даже общедоступном!) состоянии.

Цели Debian Testing и Kali Linux полностью совпадают, так что мы взяли их за основу. Чтобы добавить в дистрибутив пакеты, специфичные для Kali, мы следуем процессу, состоящему из двух этапов.

Для начала мы берем Debian Testing и принудительно внедряем наши собственные пакеты Kali (расположенные в нашем хранилище kali-dev-only) для создания репозитория kali-dev. Время от времени последний будет недоступен: например, наши пакеты, специфичные для Kali, могут не устанавливаться до тех пор, пока не будут перекомпилированы в отношении более новых библиотек. В других ситуациях пакеты, которые мы разветвляли, могут требовать обновлений, чтобы снова стать готовыми к установке или исправить проблему невозможности установки другого пакета, зависящего от более новой версии разветвленного пакета. В любом случае kali-dev не предназначен для конечных пользователей.

kali-rolling — это дистрибутив, обновления которого пользователи Kali Linux должны отслеживать и который основан на репозитории kali-dev таким же образом, как и Debian Testing базируется на Debian Unstable. Пакеты переносятся в данный дистрибутив только после проверки соответствия всех зависимостей в целевом дистрибутиве.

Управление различиями с Debian

В качестве конструктивного решения мы стараемся минимизировать количество разветвленных пакетов, насколько это возможно. Однако реализация некоторых из уникальных особенностей Kali требует внесения определенных изменений. В целях ограничения влияния этих изменений мы стремимся отправить их «выше по течению», интегрируя особенность напрямую или добавляя необходимые методы таким образом, чтобы непосредственно включить нужные функции было легко без дальнейшей модификации самих вышестоящих пакетов.

Kali Package Tracker (http://pkg.kali.org/derivative/kali-dev/) помогает нам отслеживать расхождения с Debian. В любой момент можно увидеть, какой пакет был разветвлен, синхронизирован ли он с Debian или не требуется ли ему обновление. Все наши пакеты хранятся в репозиториях Git, где рядом находятся ветки Debian и Kali. Благодаря этому обновление разветвленного пакета — простой двухэтапный процесс: обновление ветки Debian, а затем объединение ее с веткой Kali.

Хотя разветвленных пакетов в Kali сравнительно немного, количество дополнительных пакетов довольно внушительное: в апреле 2017 года их было почти 400. Большинство из них — это бесплатное ПО, соответствующее Руководству по свободному программному обеспечению Debian (Debian Free Software Guidelines) (https://www.debian.org/social_contract), и наша конечная цель заключается в том, чтобы всегда поддерживать эти пакеты в Debian. Вот почему мы стремимся следовать политике Debian (https://www.debian.org/doc/debian-policy/) и придерживаться хороших методов пакетирования, используемых в Debian. К сожалению, есть также немало исключений, когда правильное пакетирование практически невозможно. В результате нехватки времени несколько пакетов были перемещены в Debian.

1.3. Предназначение и варианты использования

Хотя основная цель Kali обобщенно может быть сформулирована как «тестирование на проникновение и аудит безопасности», за этими словами скрыто большое количество различных задач. Kali Linux создан как *фреймворк*, поскольку включает множество инструментов, предназначенных для самых различных задач (при этом они могут применяться комбинированно во время тестирования на проникновение).

К примеру, Kali Linux можно использовать на разных типах компьютеров: конечно же, на ноутбуках специалистов по тестированию на проникновение (пентестеров — penetration tester), но также и на серверах системных администраторов, желающих контролировать свою сеть, на рабочих системах криминалистических аналитиков. Что еще более неожиданно, дистрибутив можно применять на скрытых

встроенных устройствах, как правило, с процессорами ARM, которые могут работать удаленно в диапазоне беспроводной сети или быть подключенными к компьютеру целевых пользователей. Кроме того, многие ARM-устройства — прекрасные атакующие машины благодаря своим компактным размерам и небольшому количеству потребляемой энергии. Kali Linux также может работать в облаке для создания «армии» машин, занимающихся взломом паролей, и на смартфонах и планшетах, чтобы обеспечить действительно портативное тестирование на проникновение.

Но это еще не все. Пентестеры также нуждаются в серверах: чтобы задействовать программное обеспечение для совместной работы в команде, настраивать веб-сервер для использования в фишинговых кампаниях, запускать инструменты для сканирования уязвимости и для прочих соответствующих действий.

После загрузки Kali вы обнаружите, что главное меню Kali Linux организовано по темам различных задач и действий, подходящих для пентестеров и других специалистов по информационной безопасности (рис. 1.1).



Рис. 1.1. Меню приложений Kali Linux

Эти задачи и действия включают следующие.

- **Сбор информации** — сбор данных о целевой сети и ее структуре, идентификация компьютеров, их операционных систем и служб, которые они запускают. Определение потенциально уязвимых частей информационной системы. Извлечение всех видов листингов из запущенных сервисов каталогов.
- **Анализ уязвимостей** — быстрое тестирование локальной или удаленной системы на предмет подверженности влиянию ряда известных уязвимостей или ненадежных конфигураций. Сканеры уязвимостей используют базы данных, содержащие тысячи сигнатур, для выявления потенциальных уязвимостей.

- **Анализ веб-приложений** — идентификация неправильных настроек и слабых мест в безопасности веб-приложений. Крайне важно выявлять и устранять эти проблемы, учитывая, что общедоступность таких приложений делает их идеальными целями для злоумышленников.
- **Оценка базы данных** — от SQL-инъекций до атак учетных данных — атаки баз являются очень популярным направлением для злоумышленников. В меню можно найти инструменты для тестирования векторов атаки, начиная от SQL-инъекций и заканчивая извлечением и анализом данных.
- **Атаки паролей** — системы аутентификации всегда являются вектором атаки. Здесь можно найти множество полезных инструментов: от онлайн-утилит атаки паролей до автономных атак с помощью систем шифрования или хеширования.
- **Беспроводные атаки** — повсеместный характер беспроводных сетей означает, что они всегда будут популярным вектором атаки. Благодаря широкому спектру поддержки различных беспроводных карт Kali — очевидный выбор для проведения атак против множества типов беспроводных сетей.
- **Обратное проектирование** (реверс-инжиниринг) — это деятельность, включающая множество задач. В рамках поддержки атакующих действий выступает одним из основных методов выявления уязвимости и развития эксплойта. Со стороны обороны используется для анализа вредоносного ПО, применяемого в целевых атаках. В этом случае задача состоит в том, чтобы определить возможности атакующей вас шпионской программы.
- **Инструменты эксплуатации** — эксплуатация или использование уязвимости (ранее идентифицированной) позволяет получить контроль над удаленной машиной (или устройством). Данный доступ можно применить для дальнейшей эскалации атаки как на взломанном компьютере, так и на других компьютерах, доступных в его локальной сети. Эта категория содержит ряд инструментов и утилит, которые упрощают процесс создания ваших собственных эксплойтов.
- **Сниффинг и спуфинг (перехват и подмена)** — получение доступа к данным во время их перемещения по сети часто выгодно для злоумышленника. В меню вы можете найти инструменты для спуфинга, которые позволят вам выдавать себя за авторизированного пользователя, а также для сниффинга, пригодные к захвату и анализу данных прямо в момент их передачи. При совместном использовании эти инструменты могут быть весьма эффективными.
- **Пост-эксплуатация** — после получения доступа к системе вам нередко захочется поддерживать данный уровень доступа или увеличивать контроль путем продвижения по сети. В меню найдутся инструменты, которые помогут выполнить эти задачи.
- **Криминалистическая экспертиза** — криминалистическая live-среда загрузки Linux очень популярна уже много лет. Kali содержит большое количество популярных криминалистических инструментов, основанных на Linux, позволяющих выполнять все: от начальной сортировки и обработки данных до полного анализа и ведения дел.

- **Инструменты отчетности** — тестирование на проникновение завершено лишь тогда, когда был выполнен отчет о полученных результатах. Эта категория содержит инструменты, которые помогут собрать данные, полученные из средств сбора информации, обнаружить неочевидные взаимосвязи и представить объединенные сведения в различных отчетах.
- **Инструменты социальной инженерии** — когда техническая сторона хорошо защищена, часто существует возможность использования человеческого фактора в качестве вектора атаки. При грамотной мотивации людей нередко можно побудить к действиям, которые ставят под угрозу безопасность окружения. Безопасен ли PDF-файл на Flash-накопителе, который только что подключил секретарь? Или это был троянский конь, установивший бэкдор? Был ли банковский сайт, на котором только что зарегистрировался бухгалтер, оригинальным сайтом или его идеальной копией, используемой для фишинговых целей? Данная категория содержит инструменты, помогающие справляться с такими типами атак.
- **Системные сервисы** — эта категория включает средства, которые позволяют запускать и останавливать приложения, выполняемые в фоновом режиме в виде системных сервисов.

1.4. Основные характеристики Kali Linux

Kali Linux — это дистрибутив Linux, который содержит собственную коллекцию, состоящую из сотен программных средств, специально предназначенных для целевых пользователей Kali: пентестеров и других специалистов по безопасности. Он поставляется вместе с программой для установки, чтобы можно было полностью настроить Kali Linux в качестве основной операционной системы на любом компьютере.

Данный дистрибутив похож на все прочие существующие дистрибутивы Linux, но имеет и особые функции, свойственные только Kali Linux, многие из которых адаптированы к конкретным потребностям пентестеров. Рассмотрим часть этих функций.

Live-система

В отличие от большинства дистрибутивов Linux, скачиваемый ISO-образ предназначен не только для установки операционной системы; его также можно использовать в качестве самозагружаемой live-системы. Другими словами, вы можете задействовать Kali Linux без предварительной установки, просто загрузив ISO-образ (обычно после копирования образа на USB-накопитель).

Live-система содержит инструменты, наиболее часто используемые пентестерами, поэтому, даже если вы не применяете Kali Linux регулярно, то можете просто вставить диск или USB-накопитель и перезагрузиться, чтобы запустить Kali. Однако имейте в виду, что конфигурация по умолчанию не будет сохранять изменения между перезагрузками. Настройка постоянного хранилища с помощью USB-накопителя (см. раздел 9.4) позволяет задать параметры системы по своему

вкусу (например, изменять файлы конфигурации, сохранять отчеты, обновлять программное обеспечение и устанавливать дополнительные пакеты), и изменения будут сохранены при перезагрузке.

Режим криминалистической экспертизы

В основном, проводя криминалистическую работу над системой, вы пожелаете избежать любой деятельности, которая каким-либо образом способна изменить данные в анализируемой системе. К сожалению, современные рабочие среды, как правило, мешают выполнению этой задачи, пытаясь автоматически монтировать любые обнаруженные ими диски. Избежать такого поведения поможет имеющийся в Kali Linux режим криминалистической экспертизы, который можно активировать из меню загрузки: он отключит все подобные функции.

Live-система особенно полезна для криминалистических задач, поскольку позволяет перезагрузить любой компьютер в систему Kali Linux без доступа к жестким дискам или изменения данных на них.

Пользовательское ядро Linux

Kali Linux всегда предоставляет последнее настроенное ядро Linux, основанное на версии в Debian Unstable. Это обеспечивает надежную аппаратную поддержку, особенно для широкого спектра беспроводных устройств. Ядро модернизировано для поддержки беспроводной инъекции, так как многие средства оценки защищенности беспроводной сети полагаются именно на эту функцию.

Поскольку для многих аппаратных устройств требуются обновленные файлы прошивки (расположены в каталоге `/lib/firmware/`), то Kali устанавливает их по умолчанию, включая прошивку, доступную в закрытой секции Debian. Они не установлены по умолчанию в Debian, так как закрыты и, следовательно, не являются частью Debian.

Полная настраиваемость

Kali Linux создан пентестерами для пентестеров, но мы понимаем, что не все согласятся с нашими проектными решениями или выбором инструментов по умолчанию. Помня о данном обстоятельстве, мы всегда гарантируем возможность легко настроить Kali Linux в соответствии с вашими собственными потребностями и предпочтениями. С этой целью мы публикуем live-сборки, используемые для создания официальных образов Kali, чтобы вы могли настроить их по своему вкусу. Благодаря универсальности live-сборки очень просто начать работу с применения данной конфигурации и внести различные изменения, основываясь на ваших нуждах.

Live-сборка включает множество функций для модернизации установленной системы, установки вспомогательных файлов, дополнительных пакетов, выполнения произвольных команд и изменения значений, предварительно загруженных в `debconf`.

Надежная операционная система

Пользователи данного дистрибутива, имея целью обеспечение безопасности, по праву хотят знать, можно ли ему доверять и был ли он разработан таким образом, что каждый имел возможность ознакомиться с исходным кодом. Kali Linux разработан небольшой командой опытных проектировщиков, работающих прозрачно и с соблюдением лучших методов безопасности: они загружают подписанные исходные пакеты, которые затем собираются на специализированных демонах сборки. Затем пакеты проверяются и распространяются как часть подписанного репозитория.

Работа над пакетами может быть полностью просмотрена через репозитории пакетирования Git (содержащие подписанные теги), которые используются для создания исходных пакетов Kali. Кроме того, эволюцию каждого пакета можно отслеживать через контроллер пакетов Kali (http://pkg.kali.org/).

Используется на широком диапазоне ARM-устройств

Kali Linux предоставляет бинарные пакеты для таких архитектур ARM, как armel, armhf и arm64. Благодаря легко устанавливаемым образам, предоставленным Offensive Security, дистрибутив можно развернуть на различных типах устройств: от смартфонов и планшетов до Wi-Fi-роутеров и компьютеров различных форм и размеров.

1.5. Политики Kali Linux

Хотя Kali Linux стремится следовать политике Debian, насколько это возможно, все же существует ряд областей, в которых наши варианты дизайна значительно разнятся из-за специфических потребностей пентестеров.

Один суперпользователь по умолчанию

Большинство дистрибутивов Linux довольно разумно поощряют использование непривилегированной учетной записи при запуске системы и применении такой утилиты, как `sudo`, когда необходимы административные права. Это обоснованная рекомендация по безопасности, обеспечивающая дополнительный уровень защиты между пользователем и любыми потенциально деструктивными командами или операциями операционной системы. Данное замечание особенно справедливо для многопользовательских систем, где ограничение прав пользователя — обязательное требование, так как неправильное поведение одного участника может нарушить или уничтожить работу других.

Ввиду того что многие инструменты, включенные в Kali Linux, могут выполняться только с правами `root`, такая учетная запись пользователя Kali установлена по умолчанию. В отличие от других дистрибутивов Linux, вам не будет предложено

создать непривилегированного пользователя при установке Kali. Именно эта черта служит большим отличием от большинства систем Linux и, как правило, вводит в заблуждение неопытных пользователей. Новичкам нужно быть особенно осторожными в момент применения Kali, поскольку самые деструктивные ошибки возникают именно при работе с правами `root`.

Сетевые сервисы отключены по умолчанию

В отличие от Debian, в Kali Linux отключены все установленные сервисы, которые связываются по общедоступному сетевому интерфейсу по умолчанию, например HTTP и SSH.

Это решение обосновано минимизацией уязвимости во время теста на проникновение, когда небезопасно сообщать о вашем присутствии и существует риск обнаружения из-за неожиданных сетевых взаимодействий.

Вы по-прежнему можете вручную активировать любые сервисы по вашему усмотрению, запустив сервис `systemctl`. Мы вернемся к этому позже в главе 5.

Коллекция приложений с сопровождением

Debian стремится стать универсальной операционной системой и устанавливает очень мало ограничений касательно того, что будет пакетировано, при условии наличия у каждого пакета сопровождения.

Kali Linux, напротив, не пакетирует каждое средство для тестирования на проникновение. Вместо этого мы стремимся предоставить только лучшие инструменты со свободной лицензией, охватывающие большинство задач, которые могут потребоваться пентестеру.

Разработчики Kali, практикующие пентестеры, управляют процессом отбора, и мы используем их опыт и компетентность, чтобы сделать мудрый выбор. В большинстве случаев это происходит именно так, но есть и другие, более сложные решения, принятие которых просто сводится к личным предпочтениям.

Вот часть вопросов, рассматриваемых при оценке нового приложения:

- полезность приложения в контексте тестирования на проникновение;
- уникальность функций элементов приложения;
- лицензия на приложение;
- ресурсные требования приложения.

Поддержка обновленного и полезного хранилища инструментов для тестирования на проникновение — сложная задача. Мы приветствуем предложения касательно новых утилит в рамках выделенной категории (New Tool Requests — Запросы новых инструментов) в Kali Bug Tracker (https://bugs.kali.org/my_view_page.php). Мы предпочитаем получать те запросы новых программ, в которых есть подробное представление средства, в том числе объяснение его полезности, сравнение с другими подобными приложениями и т. д.

1.6. Резюме

В этой главе мы познакомили вас с Kali Linux, слегка ушли в историю, пробежались по некоторым из основных функций и показали несколько вариантов использования этого дистрибутива. Мы также обсудили ряд стратегий, которые были применены при его разработке.

- Kali Linux — это дистрибутив Linux для аудита корпоративной безопасности, основанный на Debian GNU/Linux. Kali предназначен для специалистов по безопасности и ИТ-администраторов и позволяет им проводить обширные тестирования на проникновение, криминалистический анализ и аудит безопасности.
- В отличие от большинства основных операционных систем, Kali Linux — дистрибутив с плавающим релизом, то есть *вы будете получать обновления каждый день*.
- Kali Linux основан на Debian Testing. Поэтому большинство пакетов, доступных в Kali Linux, поставляются прямо из хранилища Debian.
- Хотя основная цель Kali обобщенно может быть сформулирована как «тестирование на проникновение и аудит безопасности», существуют различные варианты использования дистрибутива, включая потребности системных администраторов, желающих контролировать свои сети, криминалистический анализ, установку встроенных устройств, беспроводной мониторинг, установку на мобильных платформах и пр.
- Меню Kali облегчает доступ к инструментам для различных задач и действий, включая анализ уязвимости, анализ веб-приложений, оценку базы данных, атаки паролей, беспроводные атаки, обратное проектирование, эксплуатационные средства, сниффинг и спуфинг, утилиты постэксплуатации, криминалистики, программы для отчетности, инструменты социальной инженерии и системные сервисы.
- Kali Linux имеет множество дополнительных функций, в том числе: использование в качестве live-системы (не предустановленной), надежный и безопасный режим криминалистики, настраиваемое ядро Linux, возможность полной настройки системы, надежная и безопасная базовая операционная система, возможность установки на ARM-устройства, безопасная сетевая политика и набор приложений с сопровождением.

В следующей главе мы перейдем к Kali Linux и опробуем его с помощью live-режима.

ГЛАВА

2

Начало работы с Kali Linux



Ключевые темы:

- скачивание ISO-образа;
- загрузка live-образа.

В отличие от некоторых других операционных систем, начать работу с Kali Linux просто благодаря тому, что его дистрибутивы представляют собой live-образы в формате ISO. Это значит следующее: вы можете загружать скачанный образ без какой-либо предварительной процедуры установки, как и применять один и тот же образ для тестирования, в качестве загрузочного образа с USB-накопителя или DVD в случае криминалистической экспертизы или для установки в качестве постоянной операционной системы на физическом или виртуальном оборудовании.

Из-за этой простоты легко забыть о необходимости соблюдать определенные меры предосторожности. Пользователи Kali часто становятся жертвами злоумышленников, будь то спонсируемые государством группы, организованные преступные группировки или отдельные хакеры. Доступ к исходному коду Kali Linux позволяет легко создавать и распространять поддельные версии, поэтому очень важно, чтобы вы завели привычку скачивать пакеты исключительно с оригинальных ресурсов и проверять целостность и подлинность скачанных файлов. Это особенно важно для специалистов по безопасности, которые часто имеют доступ к уязвимым сетям и которым доверены данные клиентов.

2.1. Скачивание ISO-образа Kali

Где скачать

Единственный официальный ресурс ISO-образов Kali Linux — раздел Downloads (Загрузки) на сайте Kali (https://www.kali.org/downloads/). Из-за популярности Kali многочисленные сайты предлагают его образы для скачивания, но их нельзя считать достоверными, и, более того, они могут быть заражены вредоносными программами или иным образом наносить непоправимый урон вашей системе.

Сайт доступен через протокол HTTPS, что затрудняет подделку файлов. Возможность выполнить атаку типа MIM (man-in-the-middle) доступна не в полном объеме, так как злоумышленнику также будет необходим сертификат https://www.kali.org/, подписанный органом сертификации Transport Layer Security (TLS), которому доверяет браузер жертвы. Поскольку органы сертификации существуют именно для предотвращения такого рода проблем, то предоставляют сертификаты только тем, чьи идентификационные данные были проверены, и тем, кто предоставил доказательства владения соответствующим сайтом.

cdimage.kali.org Ссылки, найденные на странице скачивания, указывают на домен cdimage.kali.org, который перенаправляет на ближайшее к вам зеркало, улучшая таким образом скорость передачи данных и уменьшая нагрузку на центральные серверы Kali.

Список доступных зеркал можно найти здесь: http://cdimage.kali.org/README.mirrorlist.

Что скачать

На официальной странице скачивания приведен краткий список ISO-образов (рис. 2.1).

Download Kali Linux Images

We generate fresh Kali Linux image files every few months, which we make available for download. This page provides the links to **download Kali Linux** in its latest official release. For a release history, check our Kali Linux Releases page. Please note: You can find unofficial, untested weekly releases at http://cdimage.kali.org/kali-weekly/.

Image Name	Download	Size	Version	sha256sum
Kali 64 bit	ISO | Torrent	2.6G	2017.1	49b1c5769b909220060dc4c0e11ae09d97a270a80d259e05773101df62e11e9d
Kali 32 bit	ISO | Torrent	2.7G	2017.1	501b3747e5ac7c698217392fe49ec21dacee277404500fc49d4a0ee82625aabe
Kali 64 bit Light	ISO | Torrent	0.8G	2017.1	5c0f6300bf9842b724df92cb20e4637f4561ffc03029cdcb21af3902442ae9b0
Kali 32 bit Light	ISO | Torrent	0.8G	2017.1	6c83101ecf8702c7d93d32562e822b639d5c577314b448e3b8330995e0f07e0f
Kali 64 bit e17	ISO | Torrent	2.4G	2017.1	ae293cf679f30a4f17d090a272ccb13d7619e66d4502374154106c12091fb99c
Kali 64 bit KDE	ISO | Torrent	2.7G	2017.1	839741fec378114ff068df3ec2dbed9d8e4fae613e690d50b25ce9cc1468104b
Kali 64 bit Mate	ISO | Torrent	2.6G	2017.1	3ea748aa8c5f50d80f020acdbca5f0398ee90242bb4413c12985e1865186ca9e
Kali 64 bit Xfce	ISO | Torrent	2.5G	2017.1	8a17c2f54850585760b9d32a22e26df9a28f395b401753fa0a9b298aef4c4593
Kali 64 bit LXDE	ISO | Torrent	2.5G	2017.1	35eae65aaaabba0100dfd963e45b7b4d76e0604e7721c7d232cf10320b7cae3b
Kali armhf	Image | Torrent	0.5G	2017.1	a75199aa8a3d7b64561bc03fcd6e3ff8b94743c8769eecfaa4b719f04f7cbb63
Kali armel	Image | Torrent	0.4G	2017.1	18041442219f0797c1ea5f3c18682bc4b3ced871cb3e874e90de52dd4af877c

Рис. 2.1. Список образов, предлагаемых для скачивания

Все образы дисков, помеченные как 32- или 64-разрядные, относятся к образам, подходящим для процессоров, встроенных в большинство современных стационарных и портативных компьютеров. Если вы скачиваете образ для использования на достаточно современном компьютере, то он, скорее всего, содержит 64-разрядный процессор. При наличии сомнений помните, что все 64-разрядные процессоры способны выполнять 32-разрядные инструкции. Вы всегда можете скачать и запустить 32-разрядный образ. Однако обратное утверждение неверно. Более подробную информацию вы найдете во врезке ниже.

Если вы планируете установить Kali на встроенное устройство, смартфон, Chromebook, точку доступа или любое другое устройство с процессором ARM, то должны использовать образы Linux armel или armhf.

Мой процессор 32- или 64-разрядный?

В операционной системе Windows вы можете найти эту информацию, открыв окно System (Система) на панели управления. В одноименном разделе обратите внимание на поле System Type (Тип системы): там будет указано «процессор ×64» для 64-разрядного процессора или «процессор ×86» для 32-разрядного процессора.

В операционной системе OS X/macOS нет стандартного приложения, отображающего эту информацию, но вы все же можете найти ее, введя в терминале команду uname -m. Она вернет x86_64 для системы с 64-разрядным ядром (которое может работать только на 64-разрядном процессоре), а для системы с 32-разрядным ядром команда вернет i386 или что-то подобное (i486, i586 или i686). Любое 32-разрядное приложение способно работать и на 64-разрядном процессоре, но поскольку Apple контролирует оборудование и программное обеспечение, то вряд ли вы найдете такую конфигурацию.

В Linux вы можете проверить флаги в виртуальном файле /proc/cpuinfo. Если он содержит атрибут lm, то ваш процессор 64-разрядный, в противном случае — 32-разрядный. Следующая командная строка подскажет, какой у вас процессор:

```
$ grep -qP '^flags\s*:.*\blm\b' /proc/cpuinfo && echo 64-bit
    ➥|| echo 32-bit
64-bit
```

Теперь, когда вы знаете, какой образ вам необходим — 32- или 64-разрядный, остался лишь один шаг: выбрать тип образа. Оба образа по умолчанию, как для Kali Linux, так и для Kali Linux Light, — это live-образы в формате ISO, которые можно использовать для запуска live-системы, а также для начала процесса установки. Они отличаются только набором предустановленных приложений. Образ по умолчанию поставляется с рабочим столом GNOME и большим набором пакетов, признанных подходящими для большинства пентестеров, в то время как lite-образ поставляется с рабочим столом Xfce (который менее требователен к системным ресурсам) и ограниченным набором пакетов, предоставляя возможность выбирать только необходимые приложения. Остальные образы используют альтернативные среды рабочего стола, но поставляются с тем же большим набором пакетов, что и основной образ.

После того как вы определились с необходимым образом, можете скачать его, щелкнув кнопкой мыши на ссылке ISO в соответствующей строке. Кроме того, можете скачать образ из одноранговой сети BitTorrent, нажав ссылку Torrent, при условии, что у вас есть клиент BitTorrent, предназначенный для работы с расширением `.torrent`.

Пока выбранный вами ISO-образ скачивается, вы должны обратить внимание на контрольную сумму, указанную в столбце sha256sum. Используйте ее после окончания скачивания, чтобы убедиться в соответствии скачанного образа тому, который команда разработчиков Kali разместила в сети (см. следующий подраздел).

Проверка целостности и подлинности

Специалисты по безопасности должны проверять целостность инструментов, чтобы защитить не только свои данные и сети, но также и своих клиентов. Хотя страница скачивания Kali защищена TLS, фактическая ссылка на скачивание указывает на незашифрованный URL, который не защищает от возможных MIM-атак. Факт опоры Kali на сеть внешних зеркал для распространения образа означает следующее: вы не должны слепо доверять тому, что скачиваете. Зеркало, на которое вас направили, возможно, было взломано злоумышленниками, или же вы сами можете стать жертвой атаки.

Чтобы можно было избежать этого, проект Kali всегда предоставляет контрольные суммы выпускаемых им образов. Но такая проверка будет эффективной, если вы уверены, что контрольная сумма, которую вы получили, является той же контрольной суммой, которую опубликовали разработчики Kali Linux. Существуют различные способы проверить это.

Опираясь на TLS-защищенный сайт

Когда вы получаете контрольную сумму с веб-страницы скачивания, защищенной TLS, ее происхождение косвенно подтверждается моделью безопасности сертификата X.509: видимый вами контент получен с сайта, который фактически находится под контролем человека, запросившего TLS-сертификат.

Теперь вы должны сгенерировать контрольную сумму скачанного образа и убедиться в ее соответствии сумме, которую вы получили на сайте Kali:

```
$ sha256sum kali-linux-2017.1-amd64.iso
49b1c5769b909220060dc4c0e11ae09d97a270a80d259e05773101df62e11e9d
  ➥ kali-linux-2016.2-amd64.iso
```

Если ваша сгенерированная контрольная сумма соответствует той, что находится на странице скачивания Kali Linux, то вы скачали правильный файл. При различиях в контрольной сумме возникает проблема, хотя это не говорит об угрозе или атаке; загрузки иногда повреждаются, проходя через Интернет. Повторите попытку, используя другое официальное зеркало Kali при такой возможности (дополнительную информацию о доступных зеркалах можно получить во врезке «cdimage.kali.org» (см. выше в этой главе)).

Опираясь на сеть доверия PGP

Если вы не доверяете HTTPS для аутентификации, то вы немного параноик, но это нормально. Есть много примеров плохо управляемых центров сертификации, выдавших мошеннические сертификаты, которые в конечном итоге были неправильно использованы. Вы также можете стать жертвой «дружественного» подхода MIM-атаки, внедренного во многих корпоративных сетях, если воспользуетесь специализированным хранилищем доверенных сертификатов, встроенным в браузер. Оно предоставляет поддельные сертификаты для зашифрованных сайтов, что позволяет корпоративным аудиторам контролировать зашифрованный трафик.

В таких случаях мы также предоставляем ключ GnuPG, который используем для подписи контрольных сумм поставляемых нами образов. Ниже показаны его идентификаторы:

```
pub  rsa4096/0xED444FF07D8D0BF6 2012-03-05 [SC] [expires: 2018-02-02]
     Key fingerprint = 44C6 513A 8E4F B3D3 0875 F758 ED44 4FF0 7D8D 0BF6
uid                 [ full ] Kali Linux Repository <devel@kali.org>
sub  rsa4096/0xA8373E18FC0D0DCB 2012-03-05 [E] [expires: 2018-02-02]
```

Этот ключ — часть *глобальной сети доверия*, так как был подписан хотя бы мной (Рафаэлем Херцогом), а я выступаю частью сети доверия благодаря моему интенсивному использованию GnuPG в качестве разработчика Debian.

Модель безопасности PGP/GPG уникальна. Каждый может сгенерировать любой ключ с любым удостоверением, но вы сможете доверять этому ключу лишь в том случае, если он был подписан другим ключом, которому вы уже доверяете. Когда вы подписываете ключ, вы подтверждаете, что встречали владельца ключа и знаете наверняка, что соответствующее удостоверение верно. Таким же образом вы определяете исходный набор ключей, которым вы доверяете, включая, естественно, ваш собственный ключ.

Данная модель имеет свои ограничения, поэтому у вас есть возможность скачать общедоступный ключ Kali через HTTPS (или с сервера ключей) и просто решить, что доверяете ему, поскольку его идентификатор соответствует тому, что мы указали в нескольких местах, в том числе чуть выше в этой книге:

```
$ wget -q -O — https://www.kali.org/archive-key.asc | gpg --import
[ or ]
$ gpg --keyserver hkp://keys.gnupg.net --recv-key ED444FF07D8D0BF6
gpg: key 0xED444FF07D8D0BF6: public key "Kali Linux Repository <devel@kali.org>"
     ➥imported
gpg: Total number processed: 1
gpg:               imported: 1 (RSA: 1)
[...]
$ gpg --fingerprint 7D8D0BF6
[...]
      Key fingerprint = 44C6 513A 8E4F B3D3 0875 F758 ED44 4FF0 7D8D 0BF6
[...]
```

После извлечения вы можете использовать ключ для проверки контрольных сумм предложенных образов. Скачаем файл с контрольной суммой (`SHA256SUMS`) и связанным с ним файлом подписи (`SHA256SUMS.gpg`) и проверим подпись:

```
$ wget http://cdimage.kali.org/current/SHA256SUMS
[...]
$ wget http://cdimage.kali.org/current/SHA256SUMS.gpg
[...]
$ gpg --verify SHA256SUMS.gpg SHA256SUMS
gpg: Signature made Thu 16 Mar 2017 08:55:45 AM MDT
gpg:                using RSA key ED444FF07D8D0BF6
gpg: Good signature from "Kali Linux Repository <devel@kali.org>"
```

Если вы получите сообщение `Good signature`, то можете доверять содержимому файла `SHA256SUMS` и использовать его для проверки скачанных файлов. В противном случае возникает проблема. Вы должны проверить, с официального ли зеркала Kali Linux скачали файлы.

Обратите внимание: вы можете использовать командную строку, указанную ниже, чтобы убедиться, что скачанный файл имеет ту же контрольную сумму, которая указана в `SHA256SUMS`, при условии нахождения скачанного ISO-файла в том же каталоге:

```
$ grep kali-linux-2017.1-amd64.iso SHA256SUMS | sha256sum -c
kali-linux-2017.1-amd64.iso: OK
```

Если вы не получите ответ `OK`, то скачанный файл отличается от того, который был выпущен командой Kali. Не следует доверять ему и использовать его.

Копирование образа на DVD- или USB-накопитель

Если вы не планируете запускать Kali Linux на виртуальной машине, то ISO-образ сам по себе ограничен в использовании. Вам следует записать его на DVD или скопировать на USB-накопитель, чтобы иметь возможность загрузить дистрибутив на компьютер.

Мы не будем рассказывать о том, как записать ISO-образ на DVD, поскольку процесс сильно варьируется в зависимости от платформы и среды, но в большинстве случаев после щелчка правой кнопкой мыши на файле `.iso` появится пункт контекстного меню, который запускает приложение для записи DVD. Попробуйте!

Внимание!



В этом подразделе вы узнаете, как перезаписать произвольный диск с ISO-образом Kali Linux. Всегда дважды проверяйте целевой диск перед запуском операции, так как одна ошибка, вероятно, приведет к полной потере данных и повреждению вашей установки без возможности починки.

Создание загрузочного USB-накопителя Kali в Windows

Предварительно вы должны скачать с сайта https://sourceforge.net/projects/win32diskimager/ и установить приложение Win32 Disk Imager.

Подключите USB-накопитель к своему компьютеру с установленной ОС Windows и обратите внимание на связанное с ним обозначение диска (например, `E:\`).

Запустите приложение Win32 Disk Imager и выберите ISO-файл Kali Linux, который хотите скопировать на USB-накопитель. Убедитесь, что буква выбранного устройства соответствует обозначению накопителя. Удостоверившись в выборе правильного диска, нажмите кнопку Write (Записать) и подтвердите, что хотите перезаписать содержимое USB-накопителя (рис. 2.2).

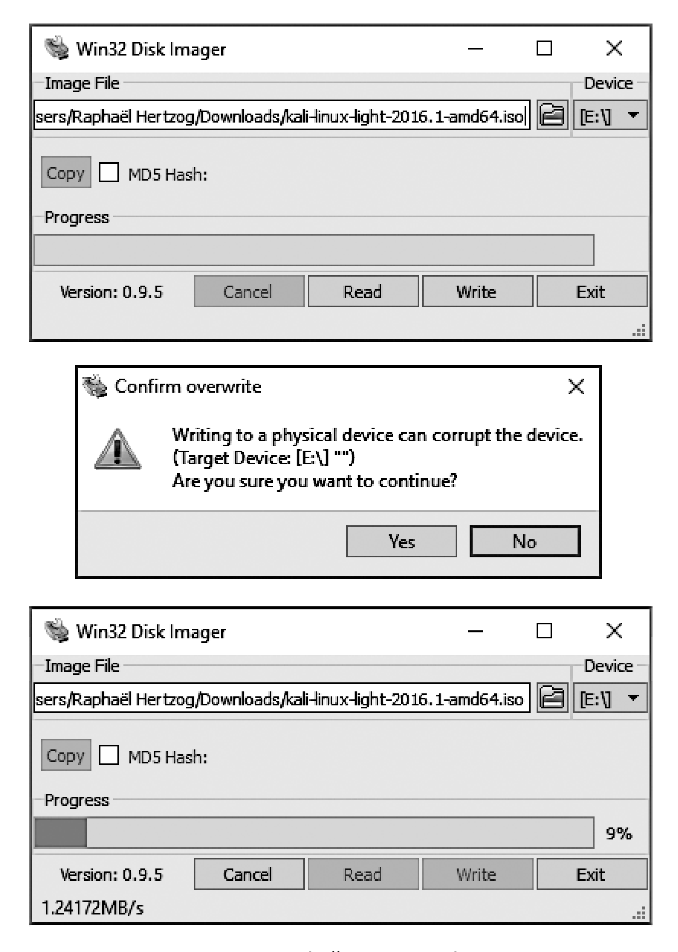


Рис. 2.2. Интерфейс Win32 Disk Imager

После завершения копирования безопасно извлеките USB-накопитель из системы Windows. Теперь вы можете использовать это USB-устройство для загрузки Kali Linux.

Создание загрузочного USB-накопителя Kali в Linux

Создать загрузочный USB-накопитель Kali Linux в среде Linux просто. Рабочая среда GNOME, установленная по умолчанию во многих дистрибутивах Linux, поставляется с утилитой Disks (в пакете gnome-disk-utility, который уже установлен в образе Kali). Эта программа показывает список дисков, обновляемый динамически при подключении или отключении носителя. Выберите свой USB-накопитель в списке дисков, и затем появится подробная информация о нем, которая поможет убедиться в выборе правильного диска. Обратите внимание, что имя устройства отображается в строке заголовка (рис. 2.3).

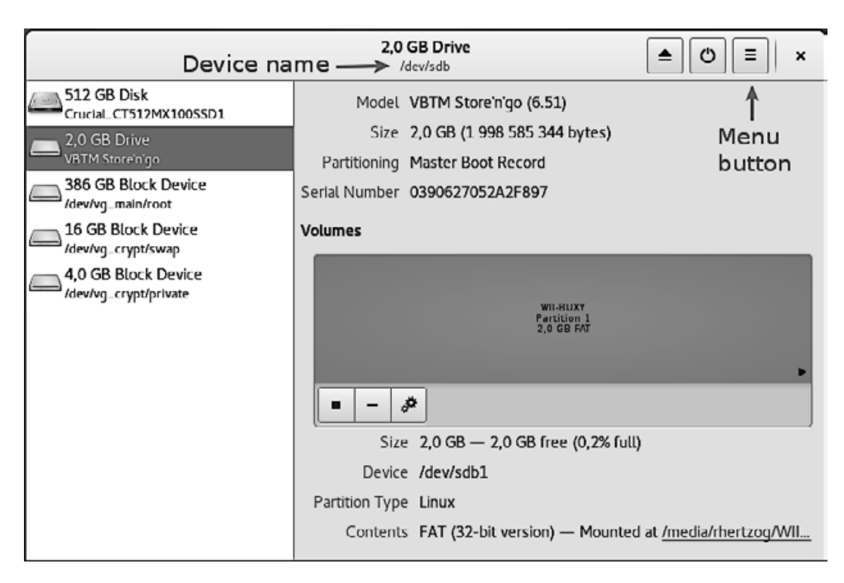


Рис. 2.3. GNOME-утилита Disks

Нажмите кнопку меню и выберите пункт Restore Disk Image (Восстановить образ диска) в появившемся меню. Выберите ISO-образ, который вы скачали ранее, и нажмите кнопку Start Restoring (Начать восстановление) (рис. 2.4).

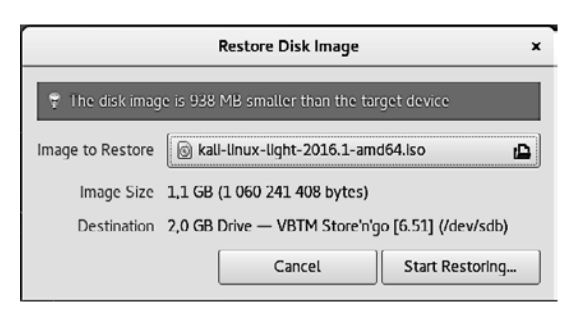


Рис. 2.4. Диалоговое окно Restore Disk Image

Насладитесь чашечкой кофе, пока он закончит копирование образа на USB-накопитель (рис. 2.5).

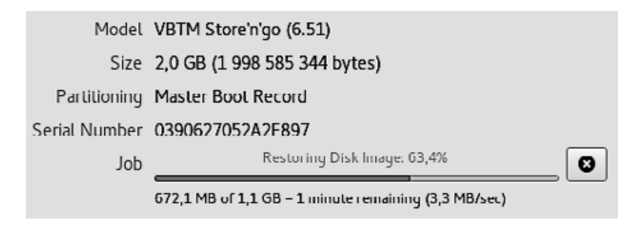


Рис. 2.5. Процесс восстановления образа

Создание загрузочного USB-накопителя из командной строки

Несмотря на то что графический процесс довольно прост, эта операция так же удобна для пользователей командной строки.

Когда вы вставляете USB-накопитель, ядро Linux обнаруживает его и назначает ему имя, которое указывается в журналах ядра. Вы можете найти его имя, проверив журналы, возвращенные командой dmesg.

```
$ dmesg
[...]
[234743.896134] usb 1-1.2: new high-speed USB device number 6
      ➥using ehci-pci
[234743.990764] usb 1-1.2: New USB device found,
      ➥idVendor=08ec, idProduct=0020
[234743.990771] usb 1-1.2: New USB device strings: Mfr=1,
      ➥Product=2, SerialNumber=3
[234743.990774] usb 1-1.2: Product: Store'n'go
[234743.990777] usb 1-1.2: Manufacturer: Verbatim
[234743.990780] usb 1-1.2: SerialNumber: 0390627052A2F897
[234743.991845] usb-storage 1-1.2:1.0: USB Mass Storage device
      ➥detected
[234743.992017] scsi host7: usb-storage 1-1.2:1.0
[234744.993818] scsi 7:0:0:0: Direct-Access VBTM Store'n'go
      ➥6.51 PQ: 0 ANSI: 0 CCS
[234744.994425] sd 7:0:0:0: Attached scsi generic sg1 type 0
[234744.995753] sd 7:0:0:0: [sdb] 3903487 512-byte logical
blocks: (2.00 GB/1.86 GiB)
[234744.996663] sd 7:0:0:0: [sdb] Write Protect is off
[234744.996669] sd 7:0:0:0: [sdb] Mode Sense: 45 00 00 08
[234744.997518] sd 7:0:0:0: [sdb] No Caching mode page found
[234744.997524] sd 7:0:0:0: [sdb] Assuming drive cache: write
      ➥through
[234745.009375] sdb: sdb1
[234745.015113] sd 7:0:0:0: [sdb] Attached SCSI removable disk
```

Теперь, когда вы знаете, что USB-накопитель доступен как устройство /dev/sdb, можете приступить к копированию образа с помощью команды dd:

```
# dd if=kali-linux-light-2017.1-amd64.iso of=/dev/sdb
2070784+0 records in
2070784+0 records out
1060241408 bytes (1.1 GB, 1011 MiB) copied, 334.175 s, 3.2 MB/s
```

Обратите внимание: для выполнения этой операции необходимы права root, и, кроме того, USB-накопитель не должен использоваться. Убедитесь в том, что ни один из его разделов не смонтирован. Команда также предполагает, что выполняется в одном каталоге с ISO-образом, иначе должен быть указан полный путь к образу.

Для справки, если указаны «исходный файл» и «файл вывода», команда dd считывает данные из исходного файла и записывает их в файл вывода. Она не отображает информацию о прогрессе, поэтому вам нужно проявить терпение, пока она выполняет свою работу (обычно для команды не требуется больше получаса!). Посмотрите

на индикатор активности записи на USB-накопителе, если хотите убедиться дважды, что команда работает. Статистика, показанная выше, отображается только после завершения команды. В OS X/macOS вы также можете нажать сочетание клавиш Ctrl+T во время операции для получения статистической информации о копировании, в том числе о том, сколько данных было скопировано.

Создание загрузочного USB-накопителя Kali в OS X/macOS

Операционная система OS X/macOS основана на UNIX, поэтому процесс создания загрузочного USB-накопителя Kali Linux похож на процедуру в Linux. После того как вы скачали и проверили выбранный ISO-файл, используйте команду `dd`, чтобы скопировать его на USB-накопитель.

Чтобы определить имя устройства USB, выполните команду `diskutil list`, и вы увидите список всех дисков, доступных в вашей системе. Затем вставьте USB-накопитель и снова запустите ту же команду. Второй вывод должен содержать дополнительный диск. Вы можете определить имя устройства USB, сравнив выводы с обеих команд. Найдите новую строку, идентифицирующую ваш USB-накопитель, и обратите внимание на `/dev/diskX`, где *X* представляет собой идентификатор диска.

Убедитесь, что USB-накопитель не смонтирован, выполнив команду демонтирования (предположительно `/dev/disk6` — это имя USB-накопителя):

```
$ diskutil unmount /dev/disk6
```

Теперь приступаем к выполнению команды `dd`. На этот раз добавьте дополнительный параметр — `bs` — для размера блока. Данный параметр определяет размер блока, который считывается из исходного файла, а затем записывается в файл вывода.

```
# dd if=kali-linux-light-2017.1-amd64.iso of=/dev/disk6 bs=1M
1011+0 records in
1011+0 records out
1060241408 bytes transferred in 327.061 secs (3242328 bytes/sec)
```

Вот и все. Ваш USB-накопитель готов, и вы можете загружать с него или использовать его для установки Kali Linux.

Загрузка с другого диска в OS X/macOS

Для загрузки с другого диска в системе OS X/macOS вызовите меню загрузки, нажав и удерживая клавишу Option сразу после включения устройства и выбора диска, который хотите использовать.

Для получения дополнительной информации см. справочную систему Apple (https://support.apple.com/ru-ru/HT202796).

2.2. Загрузка ISO-образа Kali в режиме Live

На реальном компьютере

Прежде всего вам понадобится либо подготовленный USB-накопитель (как описано в предыдущем разделе), либо DVD с записанным ISO-образом Kali Linux.

BIOS/UEFI отвечает за ранний процесс загрузки и может быть настроен через часть программного обеспечения под названием Setup. В частности, в настройках можно выбрать, какое загрузочное устройство является предпочтительным. В этом случае вы выбираете либо DVD, либо USB-накопитель, в зависимости от того, какое устройство подготовили.

Программа настройки BIOS/UEFI обычно запускается путем нажатия определенной клавиши практически сразу после включения компьютера. Этой клавишей часто является Del или Esc, а иногда F2 или F10. В большинстве случаев название нужной клавиши отображается на экране, когда компьютер включается, прежде чем загрузится операционная система.

Как только BIOS/UEFI правильно настроен для загрузки с вашего устройства, для загрузки Kali Linux остается лишь вставить DVD или подключить USB-накопитель и включить компьютер.

Отключение безопасной загрузки Хотя образы Kali Linux можно загружать в режиме UEFI, они не поддерживают безопасную загрузку. Вы должны отключить эту функцию в настройках.

В виртуальной машине

Виртуальные машины имеют множество преимуществ для пользователей Kali Linux. Эти машины особенно полезны, если вы хотите попробовать дистрибутив, но не готовы установить его на своем компьютере на постоянной основе, или если у вас мощная система и вы хотите одновременно запускать несколько операционных систем. Это популярный выбор для многих специалистов по тестированию на проникновение и профессионалов в области безопасности, которые должны задействовать широкий спектр инструментов, доступных в Kali Linux, но по-прежнему хотят иметь полный доступ к своей основной операционной системе. Это также позволяет им архивировать или безопасно удалять виртуальную машину и любые данные клиента, которые она может содержать, а не переустанавливать всю свою операционную систему.

Функции моментального снимка виртуализации программного обеспечения также позволяют легко экспериментировать с потенциально опасными операциями, такими как анализ вредоносных программ, давая возможность безболезненно выйти путем восстановления предыдущего моментального снимка.

Существует множество инструментов виртуализации, доступных для всех основных операционных систем, включая VirtualBox®, VMware Workstation®, Xen,

KVM и Hyper-V. В конечном итоге вы будете использовать тот, который вам больше всего подойдет, но мы рассмотрим два наиболее часто применяемых в настольном контексте: VirtualBox® и VMware Workstation Pro®, работающих в Windows 10. Если у вас нет ограничений корпоративной политики или личных предпочтений, то рекомендуем сначала попробовать VirtualBox, поскольку он бесплатный, хорошо работает, имеет открытый исходный код (преимущественно) и доступен для большинства операционных систем.

В следующих разделах мы предполагаем, что вы уже установили соответствующий инструмент виртуализации и знакомы с его работой.

Предварительные замечания

Чтобы воспользоваться преимуществами виртуализации в полной мере, вам следует иметь процессор с соответствующими функциями виртуализации, и они не должны быть отключены в настройках BIOS/UEFI. Дважды проверьте наличие технологии Intel® Virtualization Technology и/или Intel® VT-d Feature на экране настройки.

У вас также должна быть 64-разрядная хостовая операционная система, такая как архитектура `amd64` для дистрибутивов Linux на основе Debian, архитектура `x86_64` для дистрибутивов Linux на основе RedHat и `Windows ... 64-bit` для Windows.

Невыполнение каких-либо предварительных требований приведет к тому, что инструмент виртуализации не будет работать должным образом либо будет ограничен запуском лишь 32-разрядных гостевых операционных систем.

Поскольку инструменты виртуализации подключаются к операционной системе хоста и аппаратным средствам на низком уровне, то между ними часто возникают несовместимости. Не ожидайте, что эти инструменты станут работать хорошо одновременно. Кроме того, обратите внимание на то, что профессиональные версии Windows поставляются с установленным и включенным режимом Hyper-V, который может помешать выбранному вами инструменту виртуализации. Чтобы отключить его, выберите пункт Turn windows features on or off (Включение или отключение компонентов Windows) в настройках Windows.

VirtualBox

После первоначальной установки основное окно программы VirtualBox выглядит примерно так (рис. 2.6).

Нажмите кнопку New (Создать) (рис. 2.7), чтобы запустить мастер настройки, который поможет выполнить несколько шагов, необходимых для ввода всех параметров новой виртуальной машины.

Первым делом, как показано на рис. 2.7, вы должны присвоить имя своей новой виртуальной машине. Введите название `Kali Linux`. Вам также следует указать, какая операционная система будет применяться. Поскольку Kali Linux основан на Debian GNU/Linux, выберите тип Linux и версию Debian (32-bit) или Debian (64-bit). Хотя любая другая версия Linux, скорее всего, станет работать, это поможет внести отличие от прочих виртуальных машин, которые вы, возможно, установили ранее.



Рис. 2.6. Основное окно программы VirtualBox

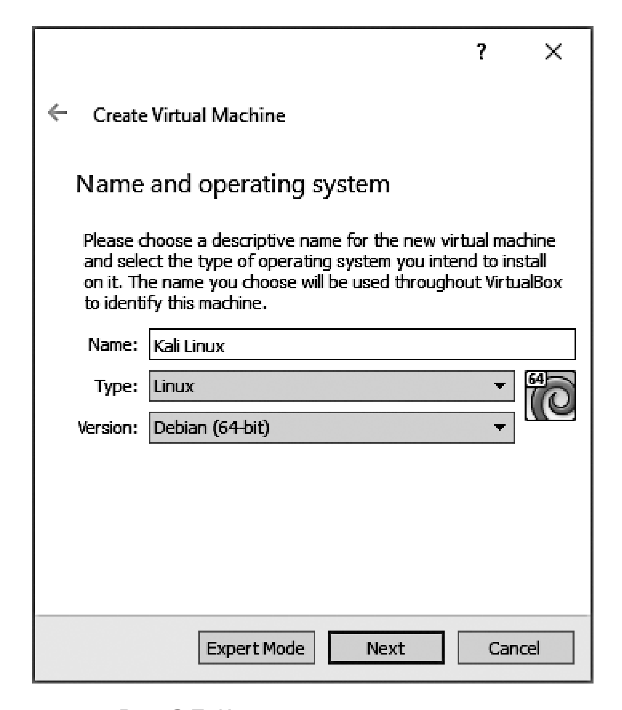


Рис. 2.7. Имя и операционная система

На втором этапе вы должны решить, сколько оперативной памяти (ОЗУ) будет выделено виртуальной машине. Хотя рекомендуемый размер 768 Мбайт приемлем для виртуальной машины Debian, выступающей в роли сервера, этого определенно недостаточно для запуска настольной системы Kali, особенно для live-системы Kali Linux, поскольку последняя использует ОЗУ для хранения изменений в файловой системе. Мы советуем увеличить значение до 1500 Мбайт (рис. 2.8) и настоятельно рекомендуем выделить не менее 2048 Мбайт ОЗУ.

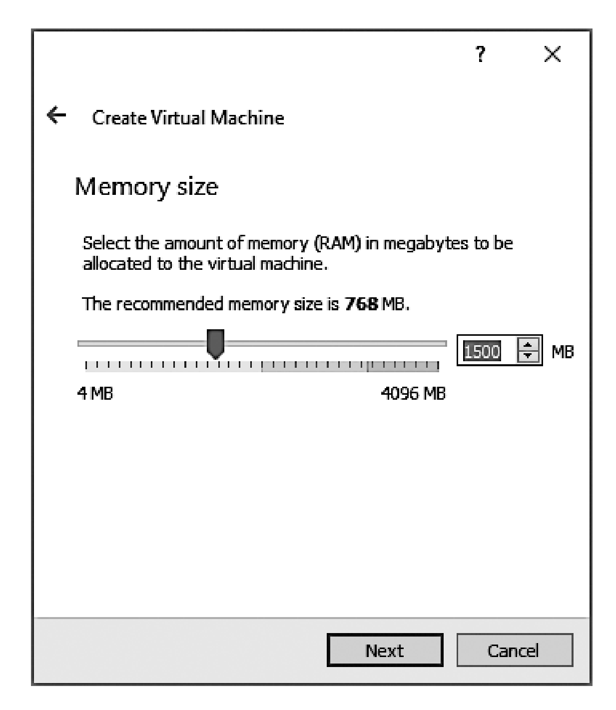


Рис. 2.8. Размер памяти

На третьем этапе (рис. 2.9) вам предлагается выбрать физический или виртуальный жесткий диск для установки виртуальной машины. Хотя жесткий диск не требуется для запуска Kali Linux в качестве live-системы, добавьте его для дальнейшей демонстрации процедуры установки (глава 4).

Содержимое жесткого диска виртуальной машины хранится на хостовой машине в виде файла. VirtualBox способен хранить это содержимое в нескольких форматах (рис. 2.10): формат по умолчанию (`VDI`) соответствует нативному формату VirtualBox; `VMDK` — формат, используемый VMware; `QCOW` — формат, применяемый QEMU. Оставьте значение по умолчанию, так как нет причин его изменять. Возможность использовать несколько форматов интересна главным образом в случае необходимости переместить виртуальную машину из одного инструмента виртуализации в другой.

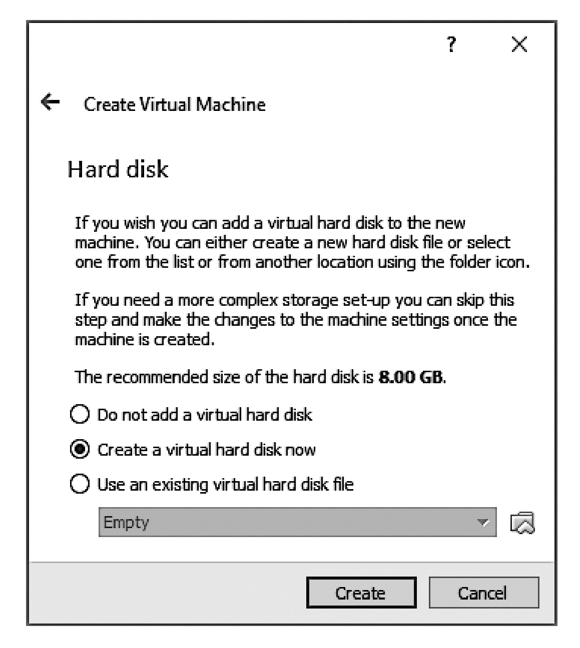


Рис. 2.9. Жесткий диск

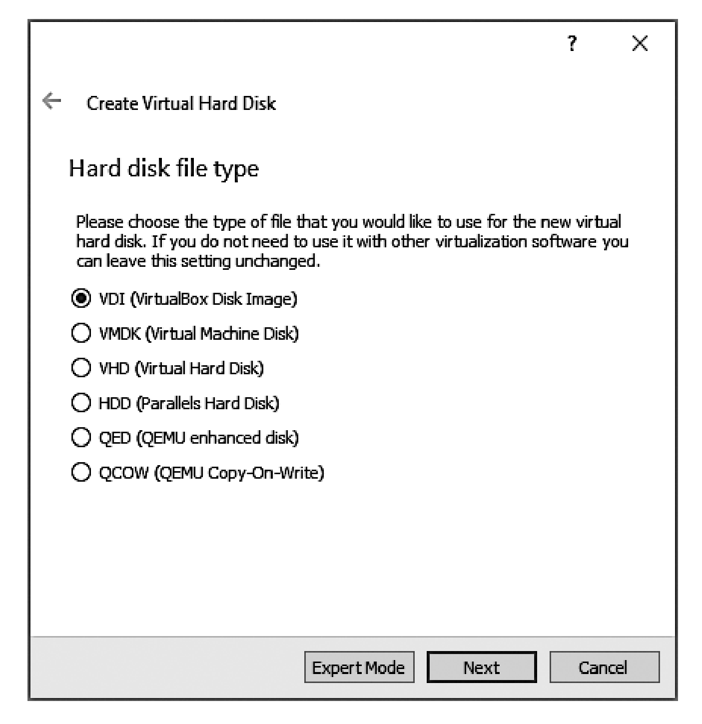


Рис. 2.10. Тип файла жесткого диска

В пояснительном тексте на рис. 2.11 четко описаны преимущества и недостатки распределения динамического и фиксированного дисков. В этом примере мы принимаем выбор по умолчанию (динамически распределенный), поскольку используем ноутбук с SSD. Мы не хотим тратить место и не нуждаемся в дополнительной производительности, так как для начала машина уже довольно быстрая.

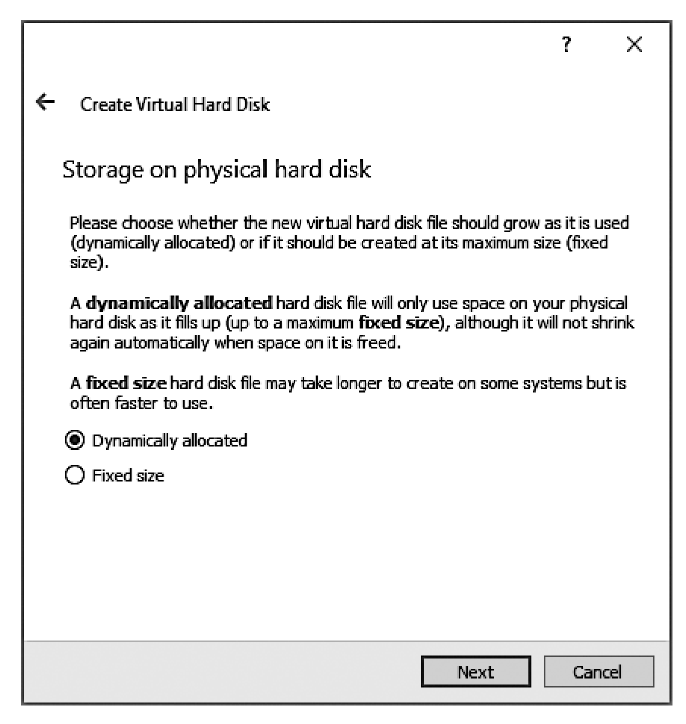


Рис. 2.11. Хранение на физическом жестком диске

Размер жесткого диска по умолчанию 8 Гбайт недостаточен для стандартной установки Kali Linux, поэтому увеличьте его до 20 Гбайт (рис. 2.12). Вы также можете указать имя и расположение образа диска. Это может быть удобно в случае, когда на жестком диске недостаточно места, поскольку позволяет сохранять образ диска на внешнем носителе.

Виртуальная машина создана (рис. 2.13), но вы еще не можете ее запустить, поскольку операционная система не установлена. Вам также следует поправить кое-какие настройки. Нажмите кнопку Settings (Настроить) на экране управления виртуальной машиной. Рассмотрим наиболее полезные из настроек.

На экране Storage (Носители) (рис. 2.14) вам следует сопоставить ISO-образ Kali Linux с виртуальным устройством чтения CD/DVD. Сначала выберите привод CD-ROM в списке дерева хранилища, а затем нажмите значок маленького компакт-диска справа, чтобы отобразить контекстное меню, в котором вы можете выбрать пункт Choose Virtual Optical Disk File (Выбрать образ оптического диска).

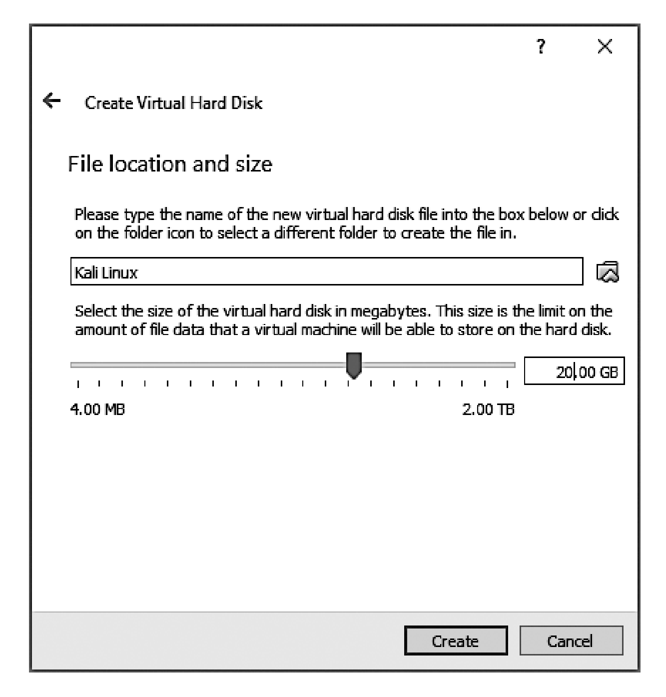


Рис. 2.12. Расположение и размер файла

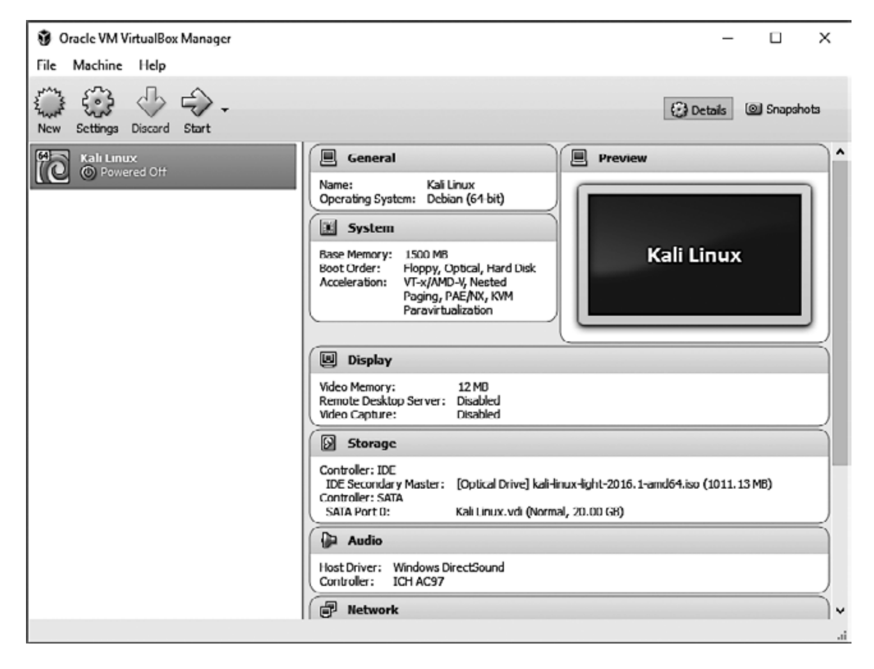


Рис. 2.13. Новая виртуальная машина в списке

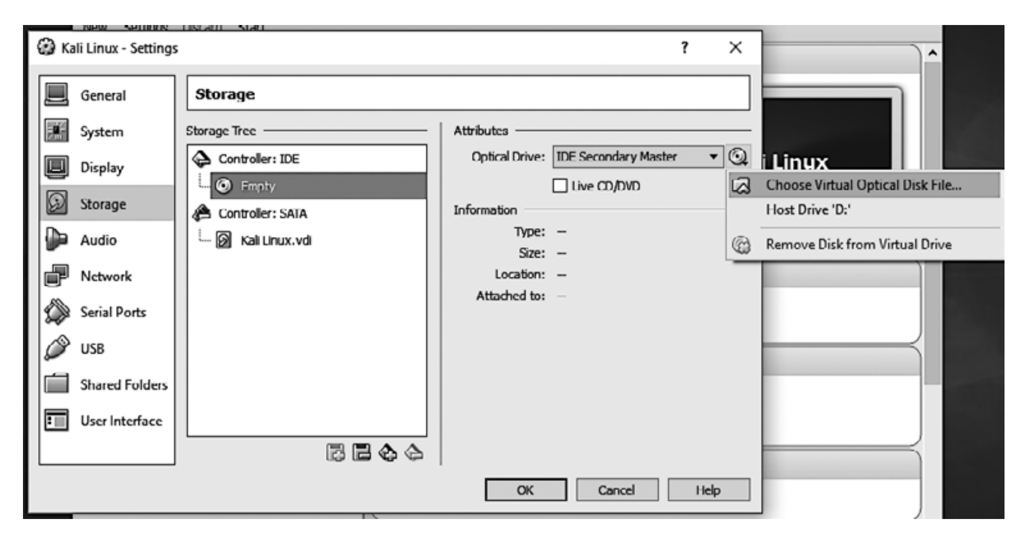


Рис. 2.14. Настройки носителей

В разделе System (Система) (рис. 2.15) вы найдете вкладку Motherboard (Материнская плата). Убедитесь, указывает ли порядок загрузки на то, что система сначала попытается загрузиться с любого оптического устройства, прежде чем перейти к жесткому диску. Кроме того, на этой вкладке вы можете изменить объем памяти, выделенный виртуальной машине, если возникнет такая необходимость.

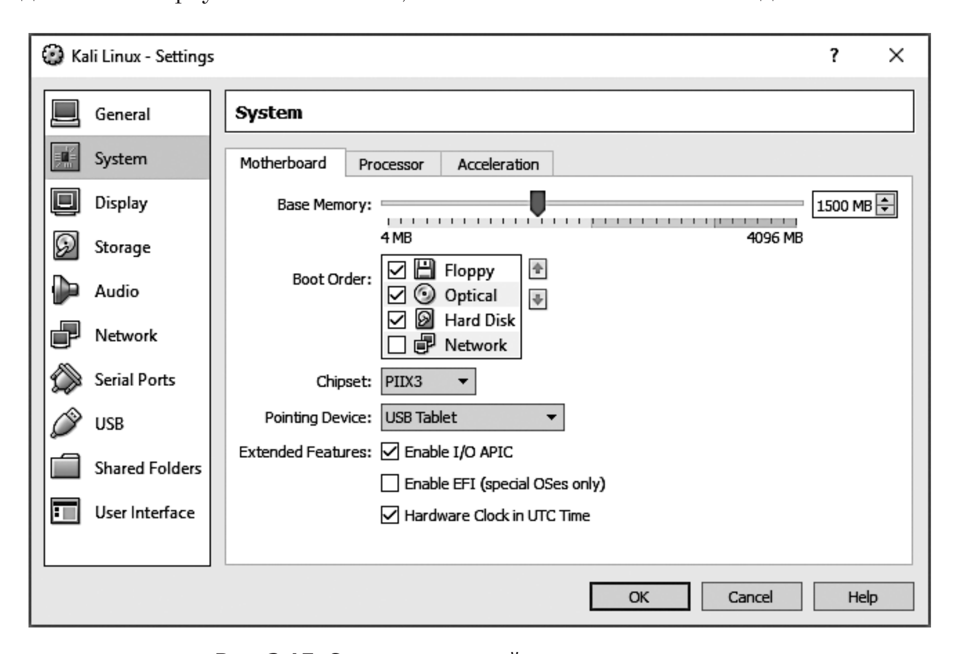


Рис. 2.15. Системные настройки: материнская плата

На этом же экране, но на вкладке Processor (Процессор) (рис. 2.16), вы можете настроить количество процессоров, назначенных виртуальной машине. Главное — при использовании 32-разрядного образа включить режим PAE/NX, иначе образ Kali не загрузится, так как вариант ядра по умолчанию, применяемый Kali для i386 (метко названный 686-pae), требует от процессора поддержки расширения физических адресов (Physical Address Extension, PAE).

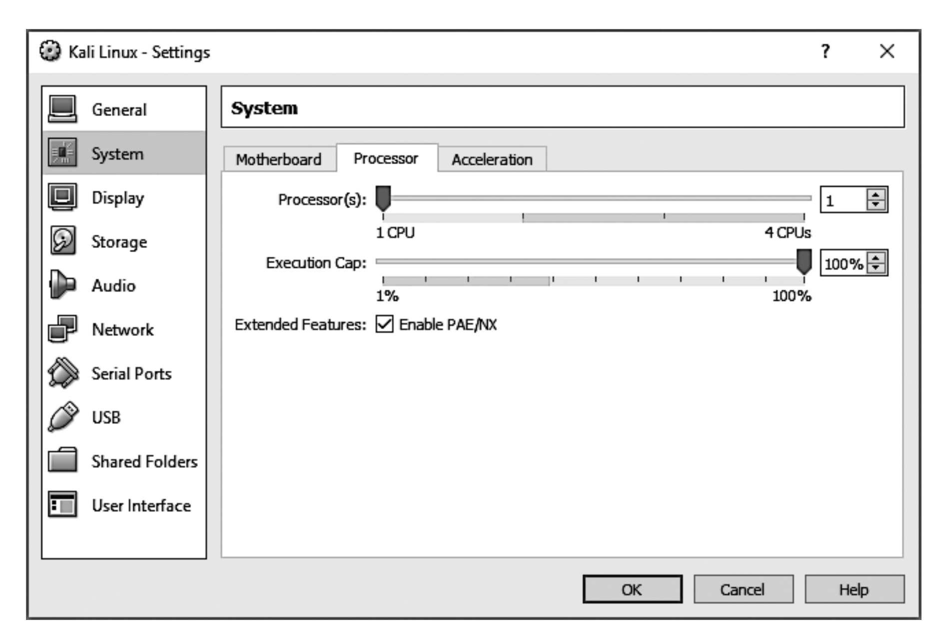


Рис. 2.16. Системные настройки: процессор

Есть много других параметров, которые можно настроить, например настройки сети (определяющие, как обрабатывается трафик на сетевой карте), но вышеупомянутых изменений достаточно, чтобы загружать рабочую live-систему Kali Linux. Наконец, нажмите кнопку Run (Запустить), и виртуальная машина должна загрузиться, как показано на рис. 2.17. Если этого не произошло, то внимательно пересмотрите все настройки и повторите попытку.

VMware

Программа VMware Workstation Pro очень похожа на VirtualBox с точки зрения функций и пользовательского интерфейса, поскольку они обе разработаны в основном для применения на стационарных компьютерах, но процесс настройки новой виртуальной машины немного отличается.

На основном экране, показанном на рис. 2.18, есть большая кнопка Create a New Virtual Machine (Создать новую виртуальную машину), запускающая мастер, который поможет создать виртуальную машину.



Рис. 2.17. Экран загрузки Kali Linux в VirtualBox



Рис. 2.18. Основной экран программы VMware

На первом этапе вы должны решить, хотите ли получить расширенные настройки во время процесса установки. В данном примере особых требований нет, поэтому выберите типичную установку (рис. 2.19).



Рис. 2.19. Мастер создания новой виртуальной машины

Мастер установки предполагает, что вы хотите установить операционную систему немедленно, и попросит выбрать ISO-образ, содержащий программу установки (рис. 2.20). Выберите пункт Installer disc image file (iso) (Файл образа диска установщика (iso)) и нажмите кнопку Browse (Обзор), чтобы выбрать файл образа.

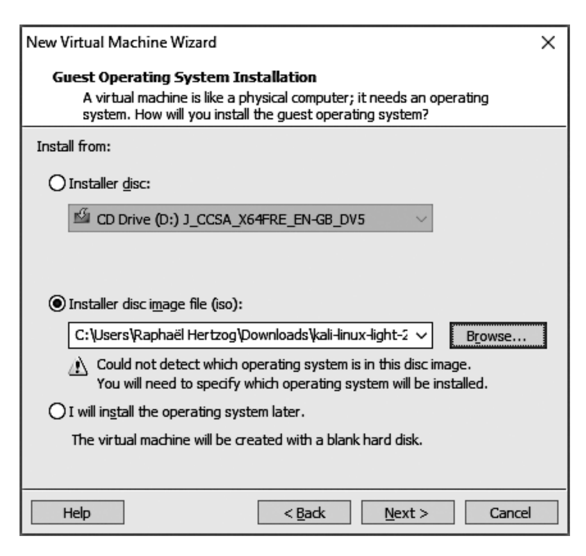


Рис. 2.20. Установка гостевой ОС

Если операционная система (ОС) не может быть определена исходя из выбранного ISO-образа, мастер спрашивает, какой тип гостевой ОС вы намереваетесь запустить. Вы должны выбрать ОС Linux и версию Debian 8.x (рис. 2.21).

Рис. 2.21. Выберите гостевую ОС

Укажите значение `Kali Linux` в качестве имени новой виртуальной машины (рис. 2.22). Как и в VirtualBox, у вас есть возможность выбрать место для хранения файлов ВМ.

Рис. 2.22. Присвойте название виртуальной машине

Размер жесткого диска по умолчанию составляет 20 Гбайт (рис. 2.23) и обычно является достаточным, но вы можете настроить его на данном этапе в зависимости от ожидаемых потребностей. В отличие от VirtualBox, который может использовать один файл разного размера, VMware позволяет хранить содержимое диска

в нескольких файлах. В обоих случаях преследуется цель сохранить дисковое пространство хоста.

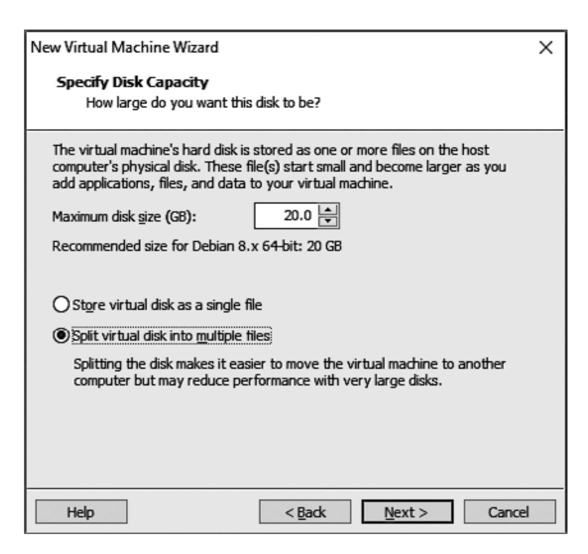


Рис. 2.23. Укажите емкость диска

Рабочая станция VMware настроена на создание новой виртуальной машины. Она отображает сводку сделанного вами выбора, позволяя проверить все до создания машины. Обратите внимание, что мастер решил выделить виртуальной машине только 512 Мбайт ОЗУ, чего недостаточно, поэтому нажмите кнопку Customize Hardware (Настроить оборудование) (рис. 2.24) и исправьте значение в разделе Memory (Память) (рис. 2.25).

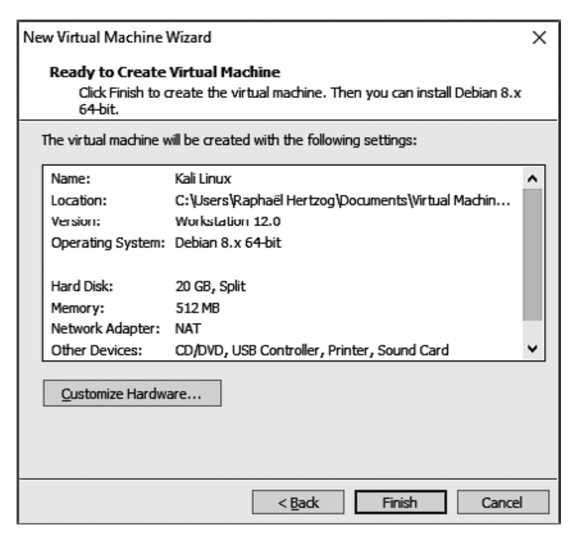


Рис. 2.24. Готово к созданию виртуальной машины

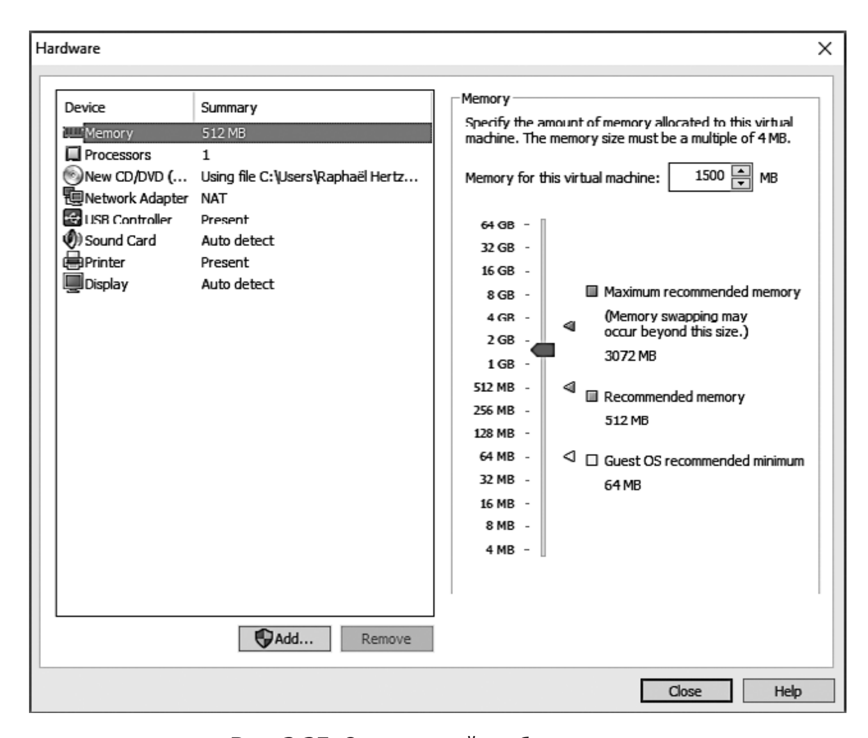


Рис. 2.25. Окно настройки оборудования

Нажмите кнопку Finish (Готово) (см. рис. 2.24). Виртуальная машина настроена и может быть запущена. Нажмите кнопку Power on this virtual machine (Включить эту виртуальную машину) (рис. 2.26).

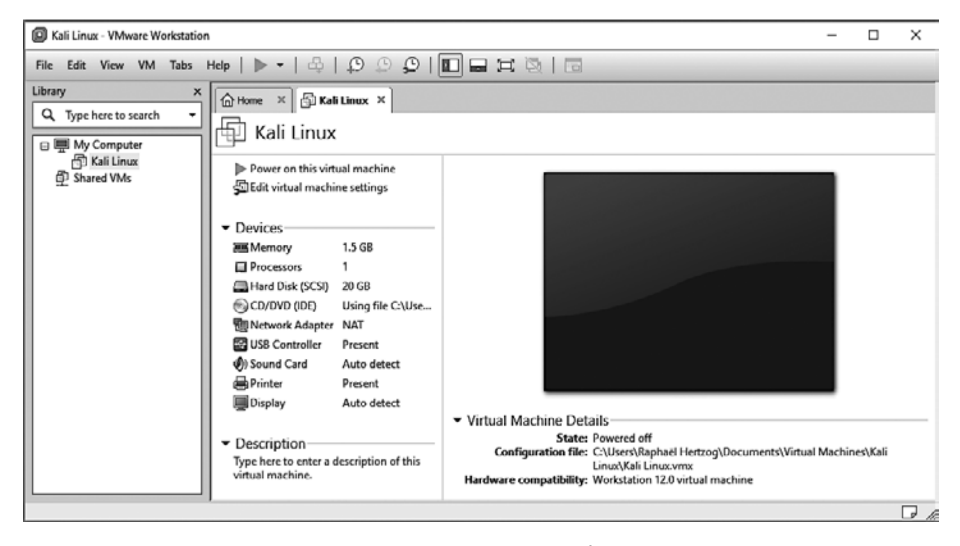


Рис. 2.26. Виртуальная машина Kali Linux готова

2.3. Резюме

В этой главе вы узнали о различных ISO-образах Kali Linux, о том, как их проверить и скачать, а также о том, как создать из них загрузочные USB-накопители в различных операционных системах. Кроме того, мы обсудили, как загружать USB-накопители, и рассмотрели, как настроить BIOS и параметры запуска на различных аппаратных платформах для их загрузки.

- Сайт https://www.kali.org/ — единственный официальный сайт для скачивания ISO Kali. Не загружайте их с других сайтов, так как эти файлы могут содержать вредоносное ПО.
- Всегда проверяйте значение `sha256sum` скачиваемых файлов одноименной командой, чтобы обеспечить целостность ISO. При выявлении несоответствий попробуйте скачать еще раз или используйте другой источник.
- Вы должны записать ISO-образ Kali Linux на загрузочный носитель, если хотите загрузить его на физическую машину. Используйте программу Win32 Disk Imager для Windows, утилиту Disks в Linux или команду `dd` в OS X/macOS. Будьте очень осторожны при записи образа. Выбор неправильного диска может навсегда повредить данные на вашем компьютере.
- Настройте BIOS/UEFI на компьютере или нажмите и удерживайте клавишу Option в OS X/macOS, чтобы машина могла загрузиться с USB-накопителя.
- Программы виртуализации, такие как VirtualBox и VMware Workstation Pro, особенно полезны, если вы хотите попробовать Kali Linux, но не готовы к его установке на постоянной основе на вашем компьютере или если у вас мощная система и вы хотите одновременно запускать несколько операционных систем.

Теперь, когда у вас есть действующая установка Kali Linux, пришло время углубиться в некоторые Linux-компоненты, необходимые для базовой и продвинутой работы Kali. Если вы уверенный или продвинутый пользователь Linux, то бегло просмотрите следующую главу.

ГЛАВА 3

Основы Linux



Ключевые темы:

- ядро Linux;
- пользовательское пространство;
- командная строка bash;
- иерархия файловой системы;
- команды Unix.

Прежде чем освоить Kali Linux, вы должны свободно себя чувствовать с системой Linux. Ее знание принесет вам немалую пользу, поскольку большой процент интернет-сервисов, таких как сайты, электронная почта и др., работают на серверах Linux.

В этой главе мы постараемся дать основы Linux, но предполагаем, что вы уже знаете о компьютерных системах в целом, включая такие компоненты, как процессор, оперативная память, материнская плата и жесткий диск, а также контроллеры устройств и связанные с ними коннекторы.

3.1. Что такое Linux и для чего она нужна

Термин Linux часто используется для обозначения всей операционной системы, но на самом деле Linux — это ее ядро, запускаемое загрузчиком, который, в свою очередь, запускается из BIOS/UEFI. Ядро исполняет роль, аналогичную роли дирижера в оркестре, обеспечивая координацию между аппаратным и программным обеспечением. Эта роль включает управление оборудованием, процессами, пользователями, разрешениями и файловой системой. Ядро обеспечивает общую базу для всех других программ в системе и обычно работает в *нулевом кольце*, также известном как *пространство ядра*.

Пользовательское пространство	Это термин для объединения всего того, что происходит за пределами ядра. Среди программ, работающих в пользовательском пространстве, много базовых утилит из проекта GNU (http://www.gnu.org/), большинство из которых предназначено для запуска из командной строки. Вы можете применять их в сценариях для автоматизации многих задач. Дополнительную информацию о наиболее важных командах вы найдете в разделе 3.4.

Кратко рассмотрим некоторые задачи, выполняемые ядром Linux.

Управление оборудованием

Ядру поручено в первую очередь управлять аппаратными компонентами компьютера. Оно обнаруживает и настраивает их при включении компьютера, а также при подключении или извлечении устройства (например USB). В результате они также становятся доступными для более высокоуровневого программного обеспечения благодаря упрощенному программному интерфейсу. Поэтому приложения могут использовать устройства, не прибегая к необходимости обращаться к деталям, скажем, к слоту расширения, в который вставлена дополнительная плата. Программный интерфейс также обеспечивает уровень абстракции; это позволяет программному обеспечению для видеоконференций, например, использовать веб-камеру независимо от ее производителя и модели. ПО может задействовать интерфейс Video

for Linux (V4L), и ядро будет переводить вызовы функций интерфейса в реальные аппаратные команды, необходимые конкретной веб-камере.

Ядро экспортирует данные об обнаруженном оборудовании через виртуальные файловые системы `/proc/` и `/sys/`. Приложения часто получают доступ к устройствам через файлы, созданные с помощью `/dev/`. Определенные файлы представляют собой диски (например, `/dev/sda`), разделы (`/dev/sda1`), мыши (`/dev/input/mouse0`), клавиатуры (`/dev/input/event0`), звуковые карты (`/dev/snd/*`), серийные порты (`/dev/ttyS*`) и другие компоненты.

Существует два типа файлов устройств: блок и символ. Первый обладает характеристиками блока данных: имеет конечный размер и позволяет получить доступ к байтам в любой позиции блока. Второй ведет себя как поток символов. Вы можете читать и писать символы, но не искать заданную позицию и изменять произвольные байты. Чтобы узнать тип файла определенного устройства, проверьте первую букву вывода команды `ls -l`. Для блочных устройств это `b`, для символьных — `c`:

```
$ ls -l /dev/sda /dev/ttyS0
brw-rw---- 1 root disk    8,  0 Mar 21 08:44 /dev/sda
crw-rw---- 1 root dialout 4, 64 Mar 30 08:59 /dev/ttyS0
```

Как и следовало ожидать, диски и разделы используют блочные устройства, тогда как мыши, клавиатуры и серийные порты — символьные. В обоих случаях программный интерфейс включает специфические для устройства команды, которые можно запустить через системный вызов `ioctl`.

Объединение файловых систем

Файловые системы — важный аспект ядра. Unix-подобные системы объединяют все хранилища файлов в одну иерархию, что позволяет пользователям и приложениям получать доступ к данным, зная их местоположение в пределах этой иерархии.

Отправная точка этого иерархического дерева — так называемый корневой каталог, обозначаемый символом `/`. Каталог может содержать именованные подкаталоги. Например, подкаталог `home` каталога `/` называется `/home/`. Этот подкаталог может, в свою очередь, содержать другие подкаталоги и т. д. Кроме того, каждый каталог может включать файлы, в которых будут храниться данные. Таким образом, `/home/buxy/Desktop/hello.txt` ссылается на файл с именем `hello.txt`, который хранится в подкаталоге `Desktop` подкаталога `buxy` каталога `home`, находящегося в корневом каталоге. Ядро переводит эту систему именования в адреса на диске и наоборот.

В отличие от других систем, Linux обладает только одной такой иерархией и может интегрировать данные с нескольких дисков. Один из этих дисков становится корневым, а остальные монтируются в каталоги в иерархии (соответствующая команда Linux называется `mount`). Эти диски затем доступны в точках монтирования. Такое положение дел позволяет хранить личные каталоги пользователей (традиционно хранятся в `/home/`) на отдельном жестком диске, который содержит каталог `buxy` (вместе с личными (домашними) каталогами других пользователей). После того как вы вмонтировали диск в `/home/`, эти каталоги будут доступны в привычном месте, а пути, такие как `/home/buxy/Desktop/hello.txt`, продолжат работать.

Существует множество форматов файловой системы, соответствующих различным способам физического хранения данных на дисках. Наиболее широко известны ext2, ext3 и ext4, но есть и другие. Например, VFAT — файловая система, которая исторически использовалась операционными системами DOS и Windows. Поддержка Linux для VFAT позволяет получить доступ к жестким дискам как под Kali, так и под Windows. В любом случае вы должны подготовить файловую систему на диске, прежде чем смонтировать ее, и эта операция называется *форматированием*.

Форматирование выполняют команды, подобные `mkfs.ext3` (где `mkfs` означает *MaKe FileSystem* — «создать файловую систему»). Эти команды требуют в качестве параметра файл устройства, представляющего раздел для форматирования (например, `/dev/sda1`, первый раздел на первом диске). Данная операция деструктивна и должна запускаться только один раз, если вы не планируете стереть файловую систему и начать работу с нуля.

Есть сетевые файловые системы, такие как NFS, которые не хранят данные на локальном диске. Вместо этого информация передается по сети на сервер, который хранит и извлекает ее по требованию. Благодаря абстракции файловой системы вам не нужно беспокоиться о том, как этот диск подключен, поскольку файлы остаются доступными по своему обычному иерархическому пути.

Управление процессами

Процесс — экземпляр программы во время выполнения, которому требуется память для хранения как самой программы, так и ее операционных данных. Ядро отвечает за создание и отслеживание процессов. Когда программа запускается, ядро сначала выделяет участок памяти, загружает в него исполняемый код из файловой системы, а затем запускает последний. Оно хранит информацию об этом процессе, наиболее явной частью которой является идентификационный номер, известный как *идентификатор процесса (PID)*.

Как и большинство современных операционных систем, системы с Unix-подобными ядрами, включая Linux, способны быть многозадачными. Другими словами, они позволяют системе одновременно запускать множество процессов. На самом деле только один процесс является запущенным в один момент времени, но ядро делит время работы процессора на небольшие фрагменты и запускает каждый процесс по очереди. Поскольку эти интервалы времени очень короткие (занимают миллисекунды), то создается впечатление, словно процессы работают параллельно, хотя на самом деле они активны только в течение их интервала и бездействуют в остальное время. Задача ядра — регулировать механизмы планирования таким образом, чтобы сохранить данный эффект, одновременно увеличивая производительность системы. Если временные интервалы будут слишком велики, то приложение может показаться не таким оперативным, как хотелось бы. Но при очень коротких интервалах система теряет время, переключая задачи слишком часто. Описанную проблему можно решить с помощью приоритетов процессов, когда высокоприоритетные процессы будут выполняться в течение более длительных периодов времени и с более частыми временными интервалами, чем процессы с низким приоритетом.

Многопроцессорные системы (и варианты)	Ограничение, приведенное выше (о выполнении только одного процесса в момент времени), справедливо не всегда: фактически оно состоит в том, что одновременно выполняется только один процесс для одного ядра процессора. Многопроцессорная, многоядерная или гиперпоточная системы позволяют работать параллельно сразу нескольким процессам. Тем не менее система разделения времени используется для обработки случаев, когда активных процессов больше, чем доступных процессорных ядер. В этом нет ничего необычного: стандартная система, даже в основном бездействующая, почти всегда имеет десятки запущенных процессов.

Ядро позволяет запускать несколько независимых экземпляров одной и той же программы, но каждому разрешается доступ только к собственным временным интервалам и памяти. Таким образом их данные остаются независимыми.

Управление правами

Unix-подобные системы поддерживают несколько пользователей и групп и позволяют управлять разрешениями. В большинстве случаев процесс идентифицируется пользователем, который его запустил. Этот процесс позволителен только для действий, разрешенных для его владельца. Так, для открытия файла требуется, чтобы ядро проверило идентификатор процесса на наличие прав доступа (более подробную информацию об этом конкретном примере см. в подразделе «Управление правами» раздела 3.4).

3.2. Командная строка

Под *командной строкой* мы подразумеваем текстовый интерфейс, который позволяет вводить команды, выполнять их и просматривать результаты. Вы можете запустить терминал (текстовый экран внутри графического рабочего стола или текстовую консоль вне любого графического интерфейса) и интерпретатор команд внутри него *(оболочку)*.

Как получить доступ к командной строке

Когда система работает исправно, самый простой способ получить доступ к командной строке — запустить терминал в графическом сеансе рабочего стола.

Например, в системе Kali Linux по умолчанию GNOME-утилиту Terminal можно запустить из списка приоритетных приложений. Вы также можете набрать слово `terminal` на экране Activities (Действия) (тот, который активируется при перемещении указателя мыши в верхний левый угол) и щелкнуть на значке соответствующего приложения (рис. 3.1).



Рис. 3.1. Запуск GNOME-утилиты Terminal

В случае неисправности графического интерфейса вы все равно можете получить доступ к командной строке в виртуальных консолях (до шести из них могут быть доступны через шесть комбинаций клавиш от Ctrl+Alt+F1 до Ctrl+Alt+F6 — клавишу Ctrl можно опустить, если вы уже находитесь в текстовом режиме, вне графического интерфейса Xorg или Wayland). Вы увидите очень простой экран входа, где нужно ввести свой логин и пароль, прежде чем получить доступ к командной строке и ее оболочке:

```
Kali GNU/Linux Rolling kali-rolling tty3
kali-rolling login: root
Password:
Last login: Fir Mar 25 12:30:05 EDT 2016 from 192.168.122.1 on pts/2
Linux kali-rolling 4.4.0-kali1-amd4 #1 SMP Debian 4.4.6-1kali1 (2016-03-18) x86_64

The programs included with the Kali GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@kali-rolling:~#
```

Программа, обрабатывающая ваш ввод и выполняющая ваши команды, называется *оболочкой* (или интерпретатором командной строки). В Kali Linux оболочкой по умолчанию является Bash (сокращенно от Bourne Again SHell). Символы `$` или `#` указывают на то, что оболочка ожидает вашего ввода. Он также сообщает, признает ли вас Bash как обычного пользователя (в первом случае (символ доллара)) или суперпользователя (второй случай (хеш)).

Основы командной строки: просмотр дерева каталогов и управление файлами

В данном подразделе представлен только краткий обзор описанных команд, каждая из которых имеет множество параметров, не указанных здесь, поэтому, пожалуйста, обратитесь к развернутой документации, доступной в соответствующих руководствах. В тестах на проникновение вы чаще будете получать доступ к системе после успешного эксплойта, а не через графический пользовательский интерфейс. Но умение работать с командной строкой важно для вашего успеха как специалиста по безопасности.

Когда сеанс начат, команда `pwd` (от print working directory — показать рабочий каталог) отображает ваше существующее местоположение в файловой системе. Текущий каталог изменяется с помощью команды `cd` *каталог* (`cd` от change directory — «сменить каталог»). Если вы не укажете целевой каталог, то будете переведены в свой личный каталог. При использовании `cd -` вы возвращаетесь в предыдущий рабочий каталог (тот, что использовался до последнего вызова команды `cd`). Родительский каталог всегда зовется `..` (две точки), тогда как текущий известен как `.` (одна точка). Команда `ls` перечисляет (listing) содержимое каталога. Если вы не указываете никаких параметров, то `ls` работает в текущем каталоге.

```
$ pwd
/home/buxy
$ cd Desktop
$ pwd
/home/buxy/Desktop
$ cd .
$ pwd
/home/buxy/Desktop
$ cd ..
$ pwd
/home/buxy
$ ls
Desktop    Downloads  Pictures  Templates
Documents  Music      Public    Videos
```

Вы можете создать новый каталог с помощью команды `mkdir` *каталог* и удалить существующий (пустой) каталог через команду `rmdir` *каталог*. Команда `mv` позволяет перемещать и переименовывать файлы и каталоги; файл удаляется

с помощью команды `rm` *файл*, а копируется благодаря команде `cp` *исходный_файл целевой_файл*.

```
$ mkdir test
$ ls
Desktop    Downloads   Pictures   Templates   Videos
Documents  Music       Public     test
$ mv test new
$ ls
Desktop    Downloads   new        Public     Videos
Documents  Music       Pictures   Templates
$ rmdir new
$ ls
Desktop    Downloads   Pictures   Templates   Videos
Documents  Music       Public
```

Оболочка выполняет команду, запустив первую программу с данным именем, которую она находит в каталоге, указанном в переменной среды `PATH`. Чаще всего эти программы находятся в каталогах `/bin`, `/sbin`, `/usr/bin` или `/usr/sbin`. Например, команда `ls` находится в каталоге `/bin/ls`; команда `which` сообщает о местоположении данного выполнения. Иногда команда напрямую обрабатывается оболочкой и в этом случае называется встроенной командой оболочки (среди них команды `cd` и `pwd`); команда `type` позволяет запросить тип каждой команды.

```
$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
$ which ls
/bin/ls
$ type rm
rm is /bin/rm
$ type cd
cd is a shell builtin
```

Обратите внимание на использование команды `echo`, которая просто отображает строку в терминале. В данном случае она служит для ввода содержимого переменной среды, поскольку оболочка автоматически заменяет переменные их значениями перед выполнением команды.

Переменные среды

Переменные среды позволяют хранить глобальные настройки для оболочки или других программ. Они контекстуальные, но наследуемые. Например, каждый процесс имеет собственный набор переменных среды (они контекстуальные). Оболочки, например входа, могут объявлять переменные, которые будут переданы другим исполняемым программам (они наследуемые).

Эти переменные можно определить общесистемно в /etc/profile или отдельно для пользователя в ~/.profile, однако не относящиеся к интерпретаторам командной строки переменные лучше помещать в /etc/environment, так как они будут введены во все пользовательские сеансы благодаря подключаемому модулю аутентификации (PAM) — даже если ни одна оболочка не выполняется.

3.3. Файловая система

Стандарт иерархии файловой системы

Как и прочие дистрибутивы Linux, Kali Linux организован в соответствии со *стандартом иерархии файловой системы* (Filesystem Hierarchy Standard, FHS), позволяя пользователям других дистрибутивов Linux легко найти нужный путь в Kali. FHS определяет назначение каждого каталога. Ниже представлены каталоги верхнего уровня:

- `/bin/` — основные программы;
- `/boot/` — ядро Kali Linux и другие файлы, необходимые на ранней стадии загрузки;
- `/dev/` — файлы устройств;
- `/etc/` — файлы конфигурации;
- `/home/` — личные файлы пользователя;
- `/lib/` — базовые библиотеки;
- `/media/*` — точки монтирования для съемных устройств (CD-ROM, USB-накопители и т. д.);
- `/mnt/` — временная точка монтирования;
- `/opt/` — дополнительные приложения, предоставляемые третьими лицами;
- `/root/` — личные файлы администратора (пользователя `root`);
- `/run/` — временные данные выполнения, которые не сохраняются при перезагрузке (еще не включены в FHS);
- `/sbin/` — системные программы;
- `/srv/` — данные, используемые серверами, размещенными в этой системе;
- `/tmp/` — временные файлы (этот каталог зачастую очищается при загрузке);
- `/usr/` — приложения (далее каталог подразделяется на `bin`, `sbin`, `lib` по той же логике, что и корневой). Кроме того, `/usr/share/` содержит независимые от архитектуры данные. Каталог `/usr/local/` предназначен для использования администратором в целях установки приложений вручную без перезаписи файлов, обрабатываемых системой пакетирования (`dpkg`);
- `/var/` — переменные данные, обрабатываемые демонами. К ним относятся файлы регистрации, очереди, буферы и кэши;
- `/proc/` и `/sys/` специфичны для ядра Linux (и не являются частью FHS). Оно использует их для экспорта данных в пространство пользователя.

Личный каталог пользователя

Содержимое этого каталога не стандартизировано, но существует несколько общепринятых норм. Первая заключается в том, что он часто обозначается тильдой (`~`). Это полезно знать, поскольку интерпретаторы команд автоматически заменяют

тильду на правильный каталог (который хранится в переменной среды `HOME` и чье обычное значение — `/home/`*`пользователь`*`/`).

Традиционно файлы конфигурации приложения часто хранятся непосредственно в личном каталоге, но имена файлов обычно начинаются с точки (например, клиент электронной почты `mutt` хранит свою конфигурацию в `~/.muttrc`). Обратите внимание: имена файлов, начинающиеся с точки, по умолчанию скрыты; команда `ls` перечисляет их только тогда, когда используется параметр `-a` и графические файловые менеджеры настроены определенным образом для отображения скрытых файлов.

Некоторые программы также используют несколько файлов конфигурации, организованных в одном каталоге (например, `~/.ssh/`). Отдельные приложения (скажем, браузер Firefox) тоже применяют свой каталог для хранения кэша скачанных данных. Это значит, что такие каталоги могут в конечном итоге потреблять много дискового пространства.

Эти файлы конфигурации, хранящиеся непосредственно в вашем личном каталоге, которые часто называются *дот-файлами*, способны распространяться до такой степени, что могут довольно сильно загромоздить каталоги. К счастью, совместная работа под эгидой FreeDesktop.org привела к созданию спецификации базового каталога XDG, соглашения, целью которого является очистка этих файлов и каталогов. В спецификации указано, что файлы конфигурации должны храниться в каталоге `~/.config`, файлы кэша — в `~/.cache`, а файлы данных приложения — в `~/.local` (или их подкаталогах). Это соглашение постепенно набирает обороты.

Графические рабочие столы обычно имеют ярлыки для отображения содержимого каталога `~/Desktop/` (`~/Рабочий стол/` или любой другой подходящий перевод для систем, настроенных не на английском языке).

И наконец, система электронной почты иногда хранит входящие письма в каталоге `~/Mail/`.

3.4. Полезные команды

Отображение и изменение текстовых файлов

Команда `cat` *`файл`* (предназначенная для связи файлов со стандартным устройством вывода) считывает файл и отображает его содержимое в терминале. Если файл слишком большой, чтобы поместиться на экране, то вы можете использовать команды пагинации (например, `less` или `more`) для отображения его постранично.

Команда `editor` запускает текстовый редактор (например, Vi или Nano) и позволяет создавать, изменять и читать текстовые файлы. Простейшие файлы иногда могут создаваться непосредственно из командной строки благодаря перенаправлению: *`команда >файл`* создает файл с именем *`файл`*, содержащим вывод данной команды. *`Команда >>файл`* действует аналогично, за исключением того, что добавляет вывод команды в файл, а не перезаписывает его.

```
$ echo "Kali rules!" > kali-rules.txt
$ cat kali-rules.txt
Kali rules!
$ echo "Kali is the best!" >> kali-rules.txt
$ cat kali-rules.txt
Kali rules!
Kali is the best!
```

Поиск файлов и по содержимому файлов

Команда `find` *каталог критерий* ищет файлы в иерархии под указанным каталогом в соответствии с некоторыми критериями. Наиболее часто используемым критерием является `-name` *имя_файла*, который позволяет искать файл по имени. Кроме того, при поиске файла по имени можно применять общие подстановочные знаки, такие как *.

```
$ find /etc -name hosts
/etc/hosts
/etc/avahi/hosts
$ find /etc -name "hosts*"
/etc/hosts
/etc/hosts.allow
/etc/hosts.deny
/etc/avahi/hosts
```

Команда `grep` *выражение файлы* выполняет поиск по содержимому файлов и извлекает строки, соответствующие регулярному выражению. Добавление параметра `-r` позволяет совершать рекурсивный поиск по всем файлам, находящимся в каталоге. Это позволяет найти файл, когда известна лишь часть его содержимого.

Управление процессами

Команда `ps aux` выводит список текущих процессов и помогает идентифицировать их, показывая их PID. Если вам известен PID процесса, команда `kill` *-сигнал pid* позволит отправить сигнал (при условии, что вы владелец процесса). Существуют различные сигналы; наиболее часто используются `TERM` (запрос на прекращение процесса) и `KILL` (принудительное завершение).

Командный интерпретатор также может запускать программы в фоновом режиме, если в конце команды указать `&`. Используя амперсанд, вы можете немедленно возобновить управление оболочкой, даже если команда все еще работает (скрыта в качестве фонового процесса). Команда `jobs` выводит список процессов, работающих в фоновом режиме; выполнение команды `fg` *%номер_задачи* (от foreground — «передний» (план)) возвращает задачу на передний план. Когда команда запускается на этом плане (была запущена в обычном режиме либо возвращена на передний план с помощью `fg`), нажатие сочетания клавиш Ctrl+Z приостанавливает процесс и возобновляет управление командной строкой. Затем данный процесс можно перезапустить в фоновом режиме через `bg` *%номер_задачи* (от background — «фоновый» (режим)).

Управление правами

Linux — многопользовательская ОС, поэтому необходима система разрешений для управления набором допустимых операций над файлами и каталогами, которая включает все системные ресурсы и устройства (в системе Unix любое устройство представлено файлом или каталогом). Данный принцип является общим для всех Unix-подобных систем.

Каждый файл или каталог имеет определенные разрешения для трех категорий пользователей:

- владелец (обозначается `u`; от user — «пользователь»);
- группа владельцев (обозначается `g`; от group — «группа»), представляющая всех членов группы;
- остальные (обозначается `o`; от other — «прочие»).

Существуют три типа прав, которые можно комбинировать:

- чтение (обозначается `r`; от read — «чтение»);
- запись (или модификация, обозначается `w`; от write — «запись»);
- запуск (обозначается `x`; от eXecute — «запуск»).

В случае с файлами эти права понятны: доступ к чтению дает возможность читать содержимое (включая копирование), доступ к записи — изменять его, а доступ к запуску позволяет его выполнять (только если это программа).

Исполняемые файлы setuid и setgid

Два особенных права относятся к исполняемым файлам: setuid и setgid (обозначаемым буквой s). Обратите внимание: мы зачастую говорим о бите, поскольку каждое из этих логических значений может быть представлено как 0 или 1. Два этих права позволяют любому пользователю запускать программу с правами владельца или группы соответственно. Данный механизм предоставляет доступ к функциям, требующим разрешения более высокого уровня, чем те, которые обычно доступны.

Поскольку программа setuid root систематически запускается под идентификатором суперпользователя, то очень важно обеспечить ее безопасность и надежность. Любой пользователь, которому удастся взломать программу setuid root для вызова команды по своему выбору, может затем действовать от имени пользователя root и иметь все права в системе. Специалисты по тестированию на проникновение всегда ищут эти файлы, когда получают доступ к системе, чтобы повысить свои привилегии.

Управление каталогом осуществляется иначе. Доступ к чтению дает право ознакомиться со списком содержимого каталога (файлов и каталогов), доступ к записи позволяет создавать или удалять файлы, а доступ к запуску — переходить через

каталог к его содержимому (например, с помощью команды `cd`). Переход через каталог без возможности прочитать его дает пользователю право доступа к тем записям в каталоге, имена которых известны.

> Безопасность
>
> **Каталог setgid и липкий бит**
>
> Бит setgid также применяется к каталогам. Любой вновь созданный элемент в таких каталогах автоматически присваивается группе владельцев родительского каталога вместо наследования основной группы создателя, как происходит обычно. Благодаря этому не нужно менять основную группу (с помощью команды newgrp) при работе в дереве файлов, совместно используемом несколькими пользователями одной определенной группы.
>
> Липкий бит (обозначаемый буквой t) — это разрешение, которое применимо только к каталогам. В частности, он используется для временных каталогов, где все имеют доступ к записи (например, /tmp/): ограничивает удаление файлов таким образом, что только их владелец или владелец родительского каталога могут их удалить. Без него любой пользователь мог бы удалять файлы других пользователей в /tmp/.

Разрешениями, связанными с файлом, управляют три команды.

- `chown` *пользователь файл* — изменяет владельца файла.

> Совет
>
> **Изменение пользователя и группы**
>
> Часто бывает необходимо одновременно изменить группу файла и его владельца. Команда chown имеет специальный синтаксис для таких случаев: chown пользователь:группа файл.

- `chgrp` *группа файл* — изменяет группу владельцев.
- `chmod` *права файл* — изменяет разрешения для файла.

Существует два способа представления прав. Их символическое представление, вероятно, проще всего понять и запомнить. В нем применяются символы, уже упомянутые выше. Вы можете определить права для каждой категории пользователей (`u/g/o`), установив их явно (с помощью `=`), добавив (`+`) или исключив (`-`). Таким образом, формула `u=rwx,g+rw,o-r` предоставляет владельцу права на чтение, запись и запуск, добавляет права на чтение и запись для группы владельцев и исключает право на чтение для остальных пользователей. Буква `a` (от all — «все») охватывает все три категории пользователей, таким образом, `a=rx` предоставляет всем трем категориям одинаковые права (чтение и запуск, но не запись).

Числовое представление (восьмеричное) связывает каждое право со значением: `4` для чтения, `2` для записи и `1` для запуска. Каждая комбинация прав связывается с суммой трех цифр, а полученное значение присваивается каждой категории пользователей в обычном порядке (владелец, группа, прочие).

Например, команда `chmod 754` *файл* устанавливает следующие права: чтение, запись и запуск для владельца (`7 = 4 + 2 + 1`); чтение и запуск для группы (`5 = 4 + 1`); только чтение для остальных. Ноль означает отсутствие прав; таким образом, `chmod 600` *файл* разрешает чтение и запись для владельца, а для других не предоставляет никаких прав. Наиболее частые комбинации прав — `755` для исполняемых файлов и каталогов и `644` для файлов данных.

Чтобы предоставить специальные права, вы можете добавить четвертую цифру по тому же принципу, где `setuid`, `setgid` и липкий бит равны `4`, `2` и `1` соответственно. Команда `chmod 4754` свяжет бит `setuid` с ранее описанными правами.

Обратите внимание: использование восьмеричной записи позволяет сразу устанавливать все права для файла; вы не можете применять ее для добавления новых прав, таких как доступ к чтению для группы владельцев, поскольку должны учитывать существующие права и вычислять новое соответствующее числовое значение.

Восьмеричная запись также используется с командой `umask`, которая позволяет ограничить разрешения на вновь созданные файлы. Когда приложение создает файл, оно назначает индикативные разрешения, зная, что система автоматически удаляет права, определенные с помощью данной команды. Введите `umask` в оболочке; вы увидите битовую маску `0022`. Это просто восьмеричное представление прав для систематического удаления (в нашем случае право на запись для группы и прочих пользователей).

Если вы зададите новое восьмеричное значение, то команда `umask` изменит маску. При использовании в файле инициализации оболочки (например, `~/.bash_profile`) она изменит маску по умолчанию для ваших рабочих сессий.

> Совет
>
> **Рекурсивная операция**
>
> Иногда нам приходится менять права для всего дерева файлов. Все вышеприведенные команды имеют параметр `-R` для рекурсивной работы в подкаталогах.
>
> Различие между каталогами и файлами иногда создает проблемы с рекурсивными операциями. Вот почему буква X была введена в символическом представлении прав. Она представляет собой право на запуск, применяемое только к каталогам (а не к файлам, не имеющим этого права). Таким образом, `chmod -R a+X` *каталог* добавляет только права запуска для всех категорий пользователей (a) для всех подкаталогов и файлов, для которых по меньшей мере одна категория пользователей (даже если у них единственный владелец) уже имеет право запуска.

Получение системной информации и файлов регистрации

Команда `free` отображает информацию о памяти; `disk free` (`df`) сообщает о доступном месте на каждом из дисков, установленных в файловой системе. Параметр `-h` (human readable — «для чтения человеком») преобразует размеры в более понятные

единицы (обычно мебибайты или гибибайты). Аналогичным же образом команда `free` поддерживает параметры `-m` и `-g` и отображает данные либо в мегабитах, либо в гибибайтах соответственно.

```
$ free
           total        used        free      shared  buff/cache      available
Mem:     2052944      661232      621208       10520      770504        1359916
Swap:          0           0           0
$ df
Filesystem    1K-blocks     Used Available Use % Mounted on
udev            1014584        0   1014584   0 % /dev
tmpfs            205296     8940    196356   5 % /run
/dev/vda1      30830588 11168116  18073328  39 % /
tmpfs           1026472      456   1026016   1 % /dev/shm
tmpfs              5120        0      5120   0 % /run/lock
tmpfs           1026472        0   1026472   0 % /sys/fs/cgroup
tmpfs            205296       36    205260   1 % /run/user/132
tmpfs            205296       24    205272   1 % /run/user/0
```

Команда `id` отображает идентификатор пользователя, выполняющего сеанс, и список групп, к которым он принадлежит. Поскольку доступ к ряду файлов или устройств может быть ограничен членами группы, то полезно будет проверить доступность группы.

```
$ id
uid=1000(buxy) gid=1000(buxy) groups=1000(buxy),27(sudo)
```

Команда `uname -a` возвращает единственную строку, в которой указаны имя ядра (Linux), имя хоста, выпуск ядра, его версия, тип машины (строка архитектуры, к примеру `x86_64`) и имя операционной системы (`GNU/Linux`). Вывод этой команды обычно должен включаться в отчеты об ошибках, так как четко определяет используемое ядро и аппаратную платформу, на которой вы работаете.

```
$ uname -a
Linux kali 4.9.0-kali3-amd64 #1 SMP Debian 4.9.18-1kali1 (2017-04-04) x86_64
    ➥GNU/Linux
```

Все эти команды предоставляют информацию о происходящем во время выполнения, но зачастую бывает необходимо проконсультироваться с файлами регистрации, чтобы понять происходившее на вашем компьютере. В частности, ядро генерирует сообщения, которые затем хранит в кольцевом буфере всякий раз, когда происходит нечто интересное (например, обнаружено новое USB-устройство, неудачное завершение операции на жестком диске или первоначальное обнаружение аппаратного обеспечения при загрузке). Вы можете извлечь файлы регистрации ядра с помощью команды `dmesg`.

Журнал системного менеджера `systemd` также хранит множество файлов регистрации (выводы демонов `stdout/stderr`, сообщения `syslog`, файлы регистрации ядра) и упрощает их получение с помощью команды `journalctl`. Если не указывать никаких аргументов, то будут просто выведены все доступные файлы регистрации в хронологическом порядке. С параметром `-r` порядок вывода изменится таким

образом, чтобы сначала отображались новые сообщения. С параметром `-f` станут непрерывно выводиться новые файлы регистрации по мере их добавления в базу данных. Параметр `-u` ограничивает вывод сообщений лишь теми, которые генерируются определенным модулем `systemd` (например, `journalctl -u ssh.service`).

Обнаружение оборудования

Ядро передает множество сведений об обнаруженном оборудовании через виртуальные файловые системы `/proc/` и `/sys/`. Определенные инструменты обобщают эти детали. Среди них `lspci` (в пакете `pciutils`) перечисляет PCI-устройства, `lsusb` (в пакете `usbutils`) — USB-устройства, а `lspcmcia` (в пакете `pcmciautils`) — карты PCMCIA. Указанные инструменты очень полезны для определения точной модели устройства. Данная идентификация также позволяет проводить более точные поисковые запросы в Интернете, что, в свою очередь, приводит к более релевантным документам. Обратите внимание: пакеты `pciutils` и `usbutils` уже установлены в базовой системе Kali, но `pcmciautils` необходимо установить с помощью команды `apt install pcmciautils`. Подробнее установку пакетов и управление ими мы рассмотрим в следующей главе. В примере 3.1 показан образец данных, предоставленных командами `lspci` и `lsusb`.

Пример 3.1. Образец информации, предоставленной командами lspci и lsusb

```
$ lspci
[...]
00:02.1 Display controller: Intel Corporation Mobile 915GM/GMS/910GML Express
   ➥Graphics Controller (rev 03)
00:1c.0 PCI bridge: Intel Corporation 82801FB/FBM/FR/FW/FRW (ICH6 Family) PCI
   ➥Express Port 1 (rev 03)
00:1d.0 USB Controller: Intel Corporation 82801FB/FBM/FR/FW/FRW (ICH6 Family)
   ➥USB UHCI #1 (rev 03)
[...]
01:00.0 Ethernet controller: Broadcom Corporation NetXtreme BCM5751 Gigabit
   ➥Ethernet PCI Express (rev 01)
02:03.0 Network controller: Intel Corporation PRO/Wireless 2200BG Network
   ➥Connection (rev 05)
$ lsusb
Bus 005 Device 004: ID 413c:a005 Dell Computer Corp.
Bus 005 Device 008: ID 413c:9001 Dell Computer Corp.
Bus 005 Device 007: ID 045e:00dd Microsoft Corp.
Bus 005 Device 006: ID 046d:c03d Logitech, Inc.
[...]
Bus 002 Device 004: ID 413c:8103 Dell Computer Corp. Wireless 350 Bluetooth
```

Эти программы имеют параметр `-v`, который предоставляет гораздо более детализированные (но обычно ненужные) данные. Наконец, команда `lsdev` (в пакете `procinfo`) перечисляет ресурсы связи, используемые устройствами.

Программа `lshw` представляет собой комбинацию указанных выше программ и отображает подробное описание аппаратного обеспечения, обнаруженного в иерархическом порядке. Вы должны прикреплять ее полный вывод к каждому отчету о проблемах поддержки данного обеспечения.

3.5. Резюме

В этой главе мы провели беглую экскурсию по структуре Linux. Мы обсудили пространства ядра и пользователя, рассмотрели многие распространенные команды оболочки Linux, процессы и способы управления ими, изучили концепции безопасности пользователей и групп, обсудили FHS и посетили некоторые из наиболее распространенных каталогов и файлов в Kali Linux.

- Linux часто используется для обозначения всей операционной системы, но на самом деле это ядро операционной системы, запускаемое загрузчиком, который, в свою очередь, запускается из BIOS/UEFI.
- Пользовательское пространство характеризует все, что происходит за пределами ядра. Среди программ, работающих в этом пространстве, есть много основных утилит из проекта GNU, большинство из которых предназначено для запуска из командной строки (текстовый интерфейс, позволяющий вводить команды, выполнять их и просматривать результат). Оболочка выполняет ваши команды через этот интерфейс.
- К основным командам относятся: `pwd` (показать рабочий каталог), `cd` (сменить каталог), `ls` (показать содержимое каталога), `mkdir` (создать каталог), `rmdir` (удалить каталог), `mv`, `rm` и `cp` (переместить, удалить или скопировать файл или каталог соответственно), `cat` (связать или показать файл), `less`/`more` (отображать файлы постранично), `editor` (запустить текстовый редактор), `find` (найти файл или каталог), `free` (отображать информацию о памяти), `df` (показать свободное место на диске), `id` (отображать идентификатор пользователя вместе со списком групп, к которым они принадлежат), `dmesg` (просмотреть файлы регистрации ядра) и `journalctl` (показать все доступные файлы регистрации).
- Вы можете проверить аппаратное обеспечение в системе Kali с помощью нескольких команд: `lspci` (список PCI-устройств), `lsusb` (список USB-устройств) и `lspcmcia` (список карт PCMCIA).
- Процесс — это экземпляр программы во время выполнения, которому требуется память для хранения как самой программы, так и ее операционных данных. Вы можете управлять процессами с помощью таких команд, как `ps` (показать процесс), `kill` (завершить процесс), `bg` (перевести процесс на задний план), `fg` (перенести фоновый процесс на передний план) и `jobs` (показать фоновые процессы).
- Unix-подобные системы являются многопользовательскими. Они поддерживают несколько пользователей и групп и позволяют контролировать действия

на основе разрешений. Вы можете управлять правами файлов и каталогов с помощью нескольких команд, включая `chmod` (изменить разрешения), `chown` (изменить владельца) и `chgrp` (изменить группу).

- Как и прочие профессиональные дистрибутивы Linux, Kali Linux организован в соответствии со стандартом иерархии файловой системы (FHS), позволяя пользователям, перешедшим из других дистрибутивов Linux, легко найти необходимый путь в Kali.
- Традиционно файлы конфигурации приложения хранятся в вашем личном каталоге, в скрытых файлах или каталогах и начинаются с периода (или точки).

Теперь, когда вы знакомы с основными принципами Linux, создадим и запустим Kali Linux.

ГЛАВА

4

Установка Kali Linux



Ключевые темы:

- установка;
- автоматическая установка;
- ARM-устройства;
- устранение неполадок.

В данной главе мы остановимся на процессе установки Kali Linux. Во-первых, обсудим минимальные требования к установке (раздел 4.1) с целью убедиться в том, что ваша реальная или виртуальная система правильно настроена для обработки выбранного вами типа установки. Затем разберем каждый шаг процесса установки (раздел 4.2) для простой установки, а также для более безопасной установки, включая полностью зашифрованную файловую систему. Кроме того, мы обсудим, как автоматизировать установку (раздел 4.3) с помощью предоставления заранее определенных ответов на вопросы установщика. Мы также продемонстрируем способы установки Kali Linux на различные ARM-устройства (раздел 4.4), что расширяет возможности Kali далеко за пределы рабочего стола. Наконец мы покажем, что делать в редком случае сбоя установки (раздел 4.5), какие действия помогут решить проблему и успешно завершить непростую установку.

4.1. Минимальные требования к установке

Требования к установке для Kali Linux различаются в зависимости от того, что вы хотите установить. По меньшей мере вы можете настроить Kali как базовый сервер Secure Shell (SSH) без рабочего стола, используя всего лишь 128 Мбайт ОЗУ (рекомендуется 512 Мбайт) и 2 Гбайт дискового пространства. Если же вы решите установить рабочий стол GNOME по умолчанию и метапакет kali-linux-full, то должны рассчитывать на как минимум 2048 Мбайт ОЗУ и 20 Гбайт дискового пространства.

Помимо требований к ОЗУ и жесткому диску ваш компьютер должен иметь процессор, поддерживаемый хотя бы одной из архитектур amd64, i386, armel, armhf или arm64.

4.2. Пошаговая установка на жесткий диск

В этом разделе мы предполагаем, что у вас есть загрузочный USB-накопитель или DVD (см. подраздел 2.1.4 «Копирование образа на DVD- или USB-накопитель» раздела 2.1 для получения подробной информации о том, как подготовить такой носитель) и вы загрузились с него, чтобы начать процесс установки.

Обычная установка

В первую очередь мы рассмотрим стандартную установку Kali с незашифрованной файловой системой.

Загрузка и запуск установщика

Как только BIOS начнет загружаться с USB-накопителя или DVD, появится меню загрузчика Isolinux (рис. 4.1). На данном этапе ядро Linux еще не загружено; меню позволяет выбрать ядро для загрузки и ввести дополнительные параметры, которые будут переданы ему в процессе.



Рис. 4.1. Экран загрузки

Для стандартной установки нужно только выбрать пункт Install (Установка) или Graphical install (Установка в графическом режиме) (с помощью клавиш со стрелками), а затем нажать клавишу Enter, чтобы запустить оставшуюся часть процесса установки.

Каждый элемент меню скрывает определенную командную строку загрузки, которая может быть настроена нажатием клавиши Tab до подтверждения ввода и загрузки.

После загрузки программа установки проведет вас шаг за шагом через весь процесс. Мы подробно рассмотрим каждый из этих шагов. Разберем установку со стандартного DVD Kali Linux; установки из образов `mini.iso` могут выглядеть несколько иначе. Мы также рассмотрим установку в графическом режиме, но единственным ее отличием от установки в классическом текстовом режиме является внешний вид. Во всех версиях задаются одинаковые вопросы и предоставляются одинаковые варианты ответов.

Выбор языка

Как показано на рис. 4.2, программа установки начинается на английском языке, однако на первом этапе вы можете выбрать язык, который будет использоваться для остальной части процесса установки. Данный выбор также послужит для определения по умолчанию наиболее подходящих вариантов в последующих этапах установки (в частности, раскладки клавиатуры).

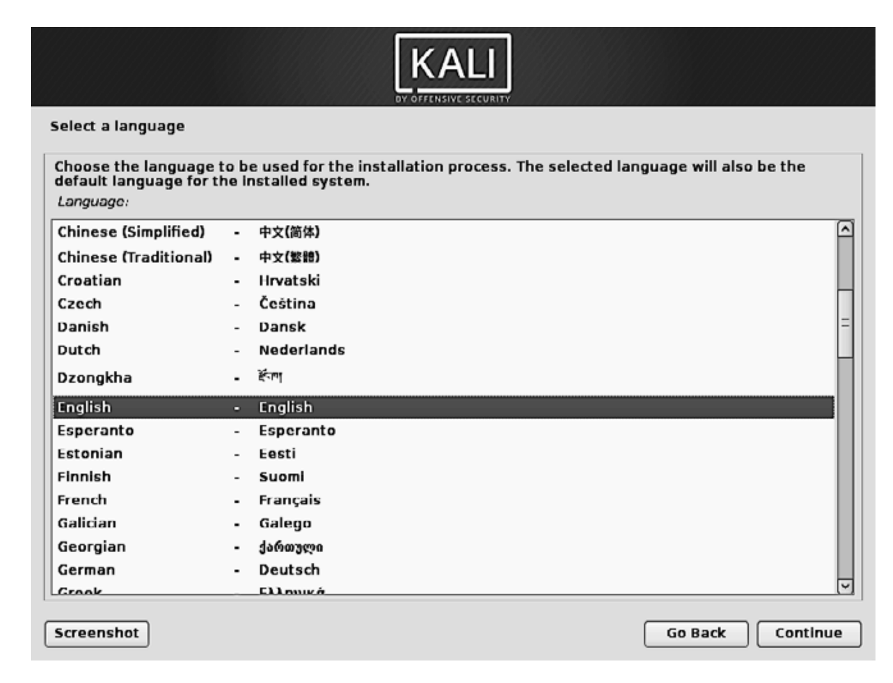


Рис. 4.2. Выбор языка

Навигация с помощью клавиатуры

Некоторые шаги в процессе установки требуют ввода информации. На экране может размещаться несколько областей, на которых необходимо будет сфокусироваться (область ввода текста, флажки, список вариантов, кнопки OK и Cancel (Отмена)), а клавиша Tab позволяет переходить от одной области к другой.

В графическом режиме установки вы можете использовать мышь, как обычно, на установленном графическом рабочем столе.

Выбор страны

Второй шаг (рис. 4.3) заключается в выборе страны. В сочетании с языком эта информация позволяет программе установки предлагать наиболее подходящую раскладку клавиатуры. Выбор страны также повлияет на настройку часового пояса. В Соединенных Штатах предлагается стандартная клавиатура QWERTY, и установщик предоставляет выбор подходящих часовых поясов.

Выбор раскладки клавиатуры

Предлагаемая американская английская клавиатура соответствует обычной раскладке QWERTY (рис. 4.4).

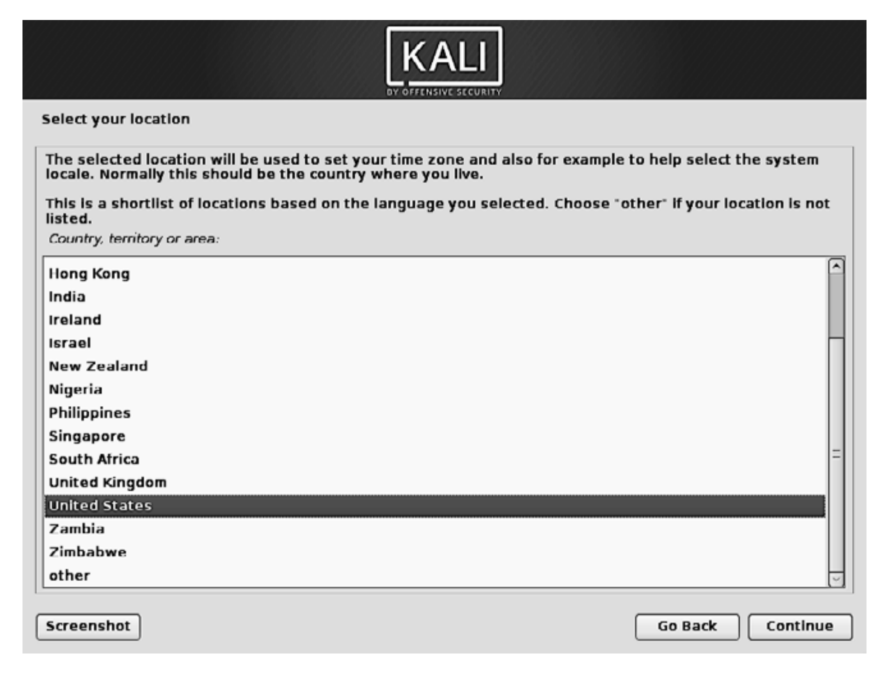


Рис. 4.3. Выбор страны

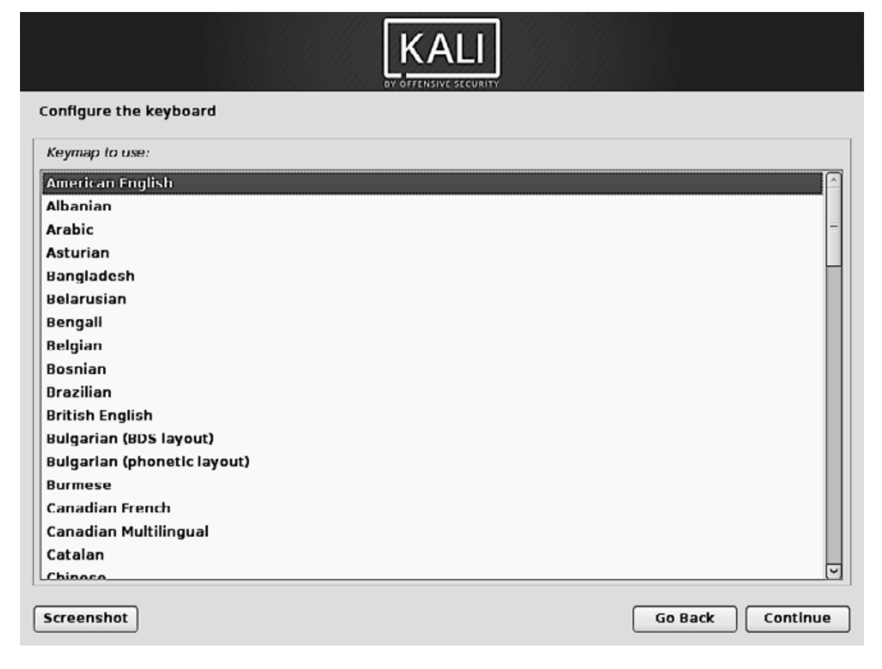


Рис. 4.4. Выбор клавиатуры

Обнаружение оборудования

В подавляющем большинстве случаев этот шаг полностью автоматизирован. Установщик обнаруживает ваше оборудование и пытается идентифицировать загрузочное устройство, чтобы получить доступ к его содержимому. Он загружает модули, соответствующие различным обнаруженным аппаратным компонентам, а затем монтирует загрузочное устройство для его чтения. Предыдущие шаги полностью содержались в загрузочном образе, имеющемся на загрузочном устройстве, файле ограниченного размера и загруженном в память загрузчиком при загрузке с загрузочного устройства.

Загрузка компонентов

Теперь, когда содержимое загрузочного устройства доступно, установщик загружает все файлы, необходимые для продолжения его работы. Сюда входят дополнительные драйверы для оставшегося оборудования (особенно для сетевой карты), а также все компоненты программы установки.

Обнаружение сетевого оборудования

На данном этапе установщик попытается автоматически идентифицировать сетевую карту и загрузить соответствующий модуль. При безуспешном завершении автоматического обнаружения вы можете вручную выбрать модуль для загрузки. Если и это не удается, то можете загрузить необходимый модуль со съемного устройства. Последнее решение обычно требуется только в том случае, когда подходящий драйвер не включен в стандартное ядро Linux, но доступен в другом месте, например на сайте производителя.

Этот шаг должен быть абсолютно успешным для сетевых установок (например, при загрузке из образов `mini.iso`), так как пакеты Debian необходимо загружать из сети.

Настройка сети

Чтобы максимально автоматизировать процесс, установщик пытается настроить автоматическую сетевую конфигурацию с помощью протокола DHCP (для IPv4 и IPv6) и протокола обнаружения соседей ICMPv6 (для IPv6) (рис. 4.5).

Если автоматическая конфигурация не удалась, то программа установки предлагает следующие варианты: повторите попытку с обычной конфигурацией DHCP, попробуйте настроить DHCP, объявив имя машины, или настройте статическую сетевую конфигурацию.

Для последней опции требуются IP-адрес, маска подсети, IP-адрес для потенциального шлюза, имя машины и имя домена.

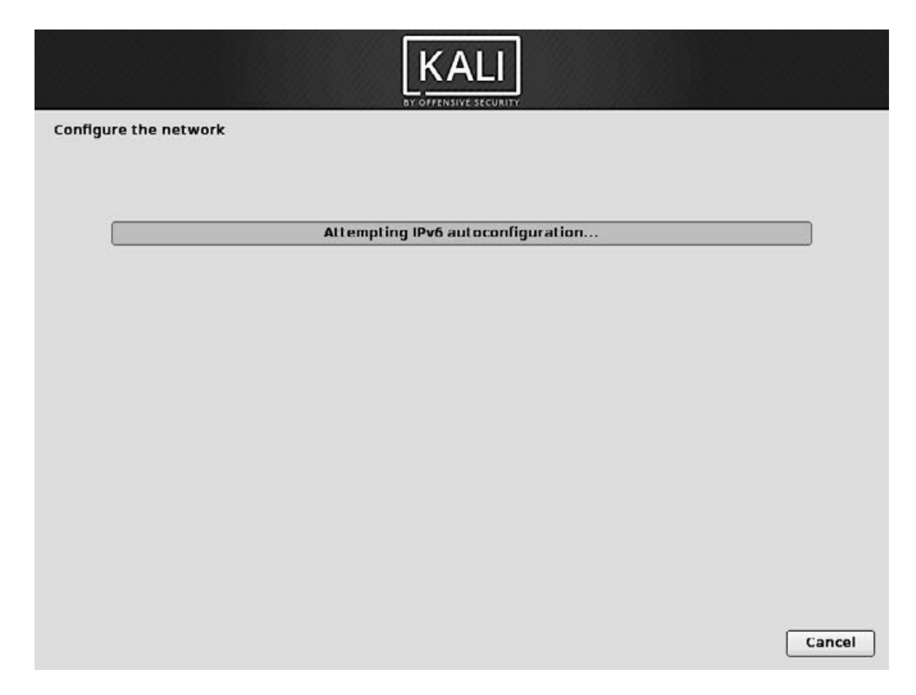


Рис. 4.5. Сетевая автоконфигурация

Конфигурация без DHCP Если локальная сеть оборудована сервером DHCP, который вы не хотите использовать, поскольку предпочитаете задать статический IP-адрес для машины во время установки, то можете добавить параметр netcfg/use_dhcp = false при загрузке. Вам просто нужно отредактировать необходимый элемент меню, нажав клавишу Tab и добавив желаемый параметр, прежде чем нажать Enter.

Пароль суперпользователя

Установщик запрашивает пароль (рис. 4.6), поскольку автоматически создает учетную запись суперпользователя типа `root`. Установщик также запрашивает подтверждение пароля, чтобы предотвратить любую ошибку ввода, которую впоследствии будет трудно устранить.

Пароль администратора Пароль пользователя root должен быть длинным (восемь символов и более) и таким, который сложно подобрать. Злоумышленники обычно нацелены на компьютеры и серверы, подключенные к Интернету,

с помощью автоматизированных инструментов, и пытаются войти в систему с очевидными паролями. Иногда злоумышленники применяют словарные атаки, используя множество комбинаций слов и чисел в качестве паролей. Старайтесь не указывать имена детей или родителей и даты рождения, потому что их легко угадать.

Эти замечания одинаково актуальны для других паролей пользователей, но последствия скомпрометированной учетной записи менее критичны для пользователей без административных прав.

Если вам не хватает вдохновения, не стесняйтесь использовать генератор паролей, такой как pwgen (найденный в пакете с тем же именем, который уже включен в базовую установку Kali).

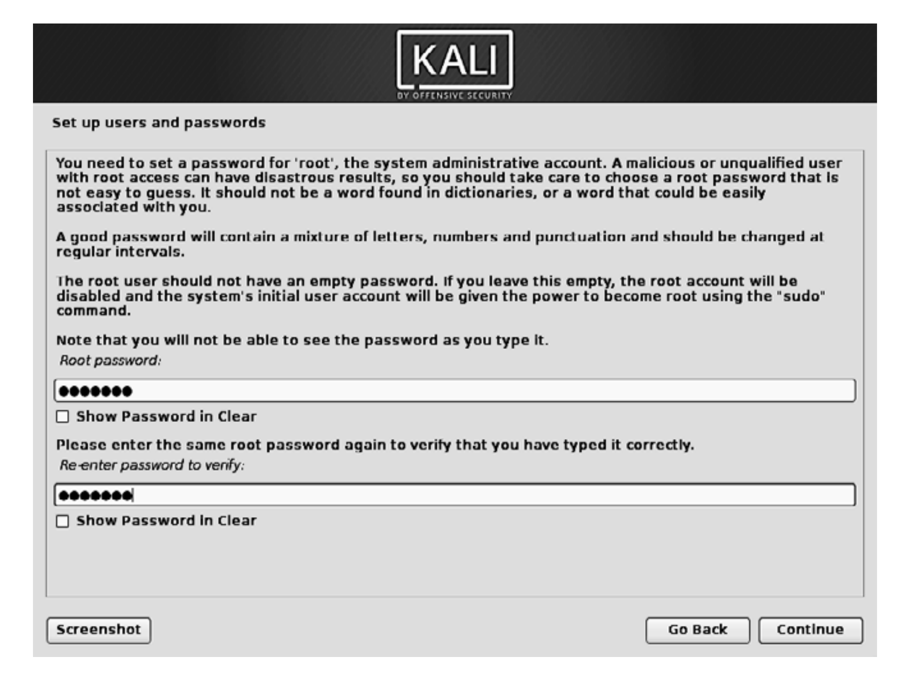


Рис. 4.6. Пароль суперпользователя

Настройка времени

В случае доступности сети внутренние часы системы будут обновляться с сервера протокола времени сети (NTP). Это целесообразно, поскольку гарантирует корректность отметок времени в журналах с первой загрузки.

Если ваша страна охватывает несколько часовых поясов, то вам будет предложено выбрать тот из них, который вы хотите использовать (рис. 4.7).

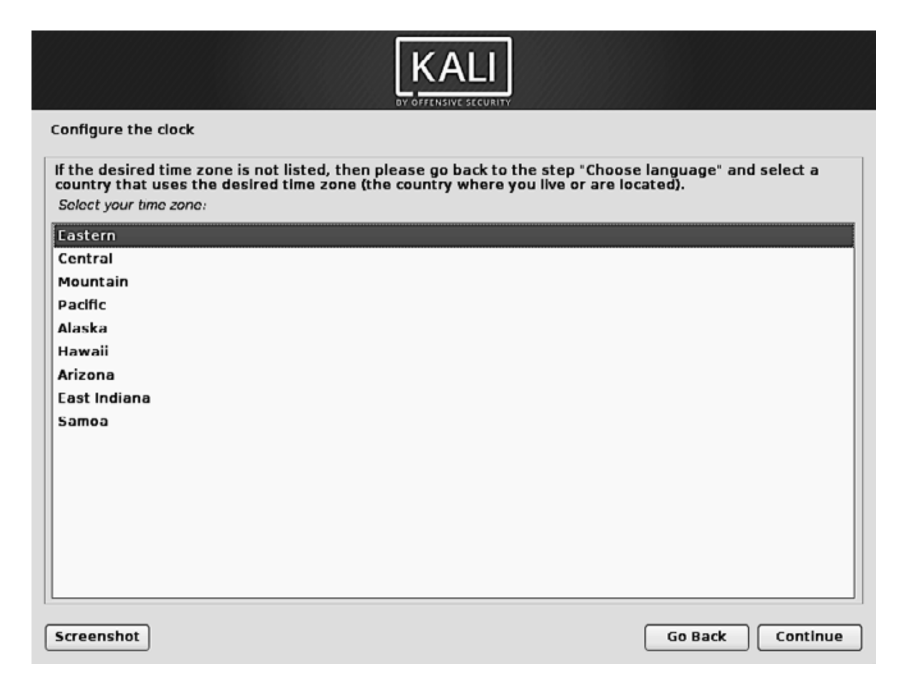


Рис. 4.7. Выбор часового пояса

Обнаружение дисков и других устройств

На данном этапе автоматически определяются жесткие диски, на которых может быть установлен Kali, и каждый из них будет рассмотрен на следующем шаге.

Выделение разделов

Выделение разделов (разметка) — неотъемлемый шаг в установке, состоящий в том, чтобы разбить доступное пространство на жестких дисках на отдельные *секции (разделы)* в соответствии с предполагаемой функцией компьютера и этих разделов. Разметка также предполагает выбор файловых систем, которые будут использоваться. Все эти решения окажут влияние на производительность, безопасность данных и администрирование серверов.

Этап разметки традиционно сложен для новых пользователей. Однако файловые системы и разделы Linux, включая виртуальную память (или разделы *подкачки*), должны быть определены, поскольку образуют основу системы. Данная задача может усложниться, если вы уже установили на компьютере другую операционную систему и хотите, чтобы обе они сосуществовали. В таком случае вы должны быть уверены, что не будете изменять их разделы или, если потребуется, измените их размер, не причинив ущерба.

Для применения более распространенных (и более простых) схем разделов большинство пользователей предпочтет *управляемый* режим (Guided), который рекомендует настройки разделов и предлагает рекомендации на каждом шагу. Более продвинутые пользователи оценят *ручной* режим (Manual), позволяющий задать дополнительные настройки. Каждый режим имеет определенные возможности.

Управляемая разметка. Первый экран в инструменте разметки (рис. 4.8) представляет точки входа для управляемого и ручного режимов разметки. Guided — use whole disk (Управляемый — использовать весь диск) — это самая простая и наиболее распространенная схема разметки, которая выделяет весь диск для Kali Linux.

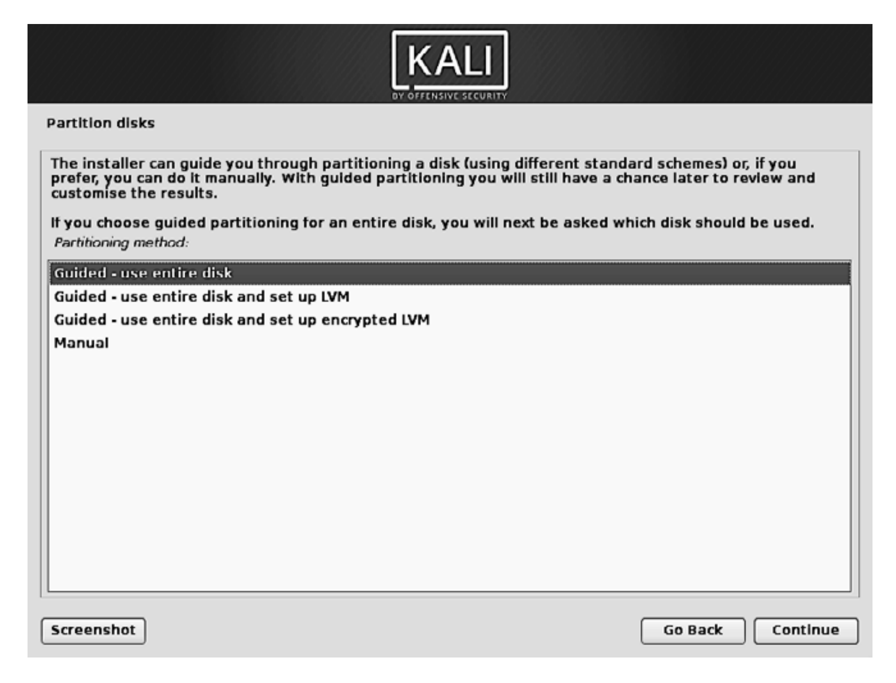


Рис. 4.8. Выбор режима разметки

Два следующих варианта используют инструмент Logical Volume Manager (LVM) для настройки логических (а не физических), при необходимости зашифрованных разделов. Ниже мы обсудим LVM и шифрование.

Наконец, последний вариант инициирует ручную разметку, которая позволяет использовать более продвинутые схемы разметки, такие как установка Kali Linux наряду с другими операционными системами. Мы обсудим ручной режим чуть позже.

В данном примере мы выделим весь жесткий диск для Kali, поэтому выбираем вариант Guided — use whole disk (Управляемый — использовать весь диск), чтобы перейти далее.

На следующем этапе (рис. 4.9) можно выбрать диск, на котором будет установлен Kali, указав соответствующий вариант (например, Virtual disk 1 (vda) — 32.2 GB Virtio Block Device). После выбора управляемая разметка продолжится. Опция удалит все данные на этом диске, так что выбирайте обдуманно.

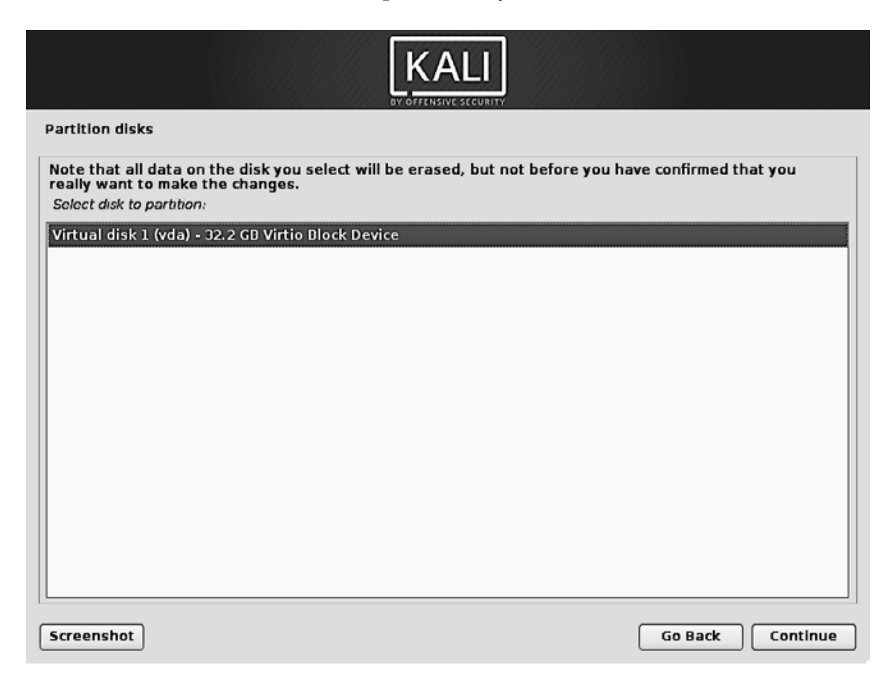


Рис. 4.9. Диск для использования

Далее инструмент управляемой разметки предлагает три метода разметки, которые соответствуют различным способам применения (рис. 4.10).

Первый метод называется All files in one partition (Все файлы в одном разделе). Дерево системы Linux хранится в единой файловой системе, соответствующей корневому каталогу (/). Эта простая и надежная схема разметки отлично работает для персональных или однопользовательских систем. Несмотря на название, на самом деле создадутся два раздела: в первом будет размещена полная система, во втором — виртуальная память.

Второй метод — Separate /home/ partition (Отделить раздел /home) — аналогичен, но разделяет иерархию файлов на две части: один раздел содержит систему Linux (/), а второй — личные каталоги (что означает доступ к данным пользователя в файлах и подкаталогах через /home/). Одно из преимуществ этого метода заключается в том, что при переустановке системы можно будет легко сохранить данные пользователя.

Последний метод разметки под названием Separate /home, /var, and /tmp partitions (Отделить разделы /home, /var и /tmp) подходит для серверов и многопользовательских

систем. Он делит дерево файлов на множество разделов: помимо `root` (`/`) и учетных записей пользователей (`/home/`) он также имеет разделы для данных программного обеспечения сервера (`/var/`) и временных файлов (`/tmp/`). Одно из преимуществ этого метода заключается в том, что конечным пользователям не под силу блокировать сервер, потребляя все доступное пространство на жестком диске (они могут заполнять только `/tmp/` и `/home/`). В то же время данные демона (особенно журналы) больше не смогут забивать остальную часть системы.

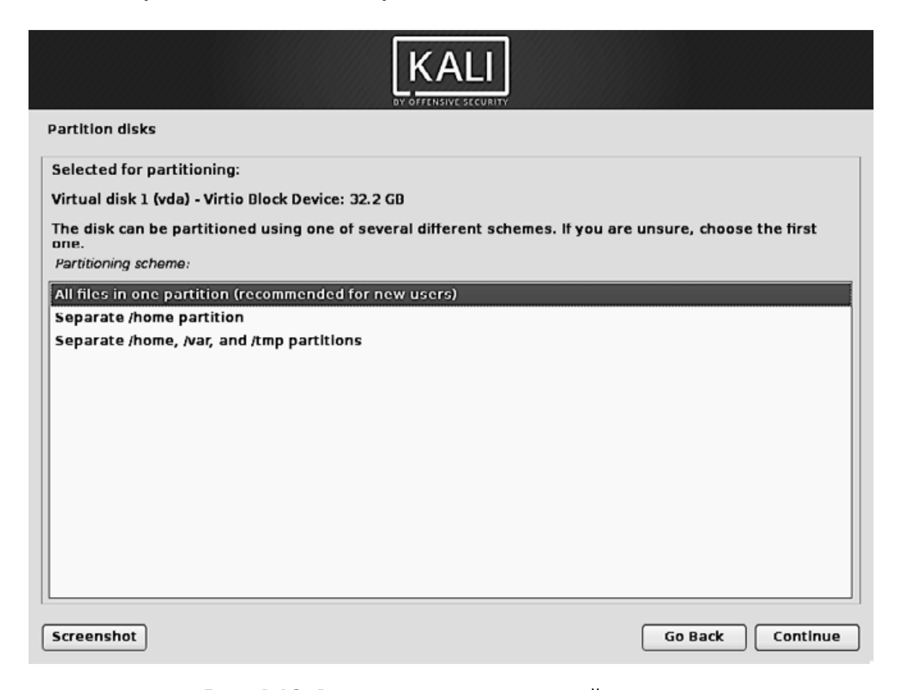


Рис. 4.10. Распределение управляемой разметки

После того как вы определили тип разметки, установщик представит сводку ваших выборов на экране в виде карты разделов (рис. 4.11). Вы можете изменить каждый раздел отдельно, выделив его, например, выбрать другую файловую систему, если стандартная (ext4) не подходит. Однако в большинстве случаев предлагаемая разметка является разумной, и вы можете принять ее, выбрав пункт Finish partitioning and write changes to disk (Завершить разметку и записать изменения на диск). Само собой разумеется, что выбирать нужно с умом, поскольку это приведет к стиранию содержимого выбранного диска.

Ручная разметка. Выбор ручного режима на главном экране Partition disks (Диски разделов) (см. рис. 4.8) обеспечивает большую гибкость, позволяя выбирать более сложные конфигурации и конкретно диктовать назначение и размер каждого раздела. Например, в этом режиме вы можете установить Kali вместе с другими операционными системами, включить резервный массив независимых дисков (RAID)

на базе программного обеспечения для защиты данных от сбоев жесткого диска, безопасно изменять размеры существующих разделов без потери данных и пр.

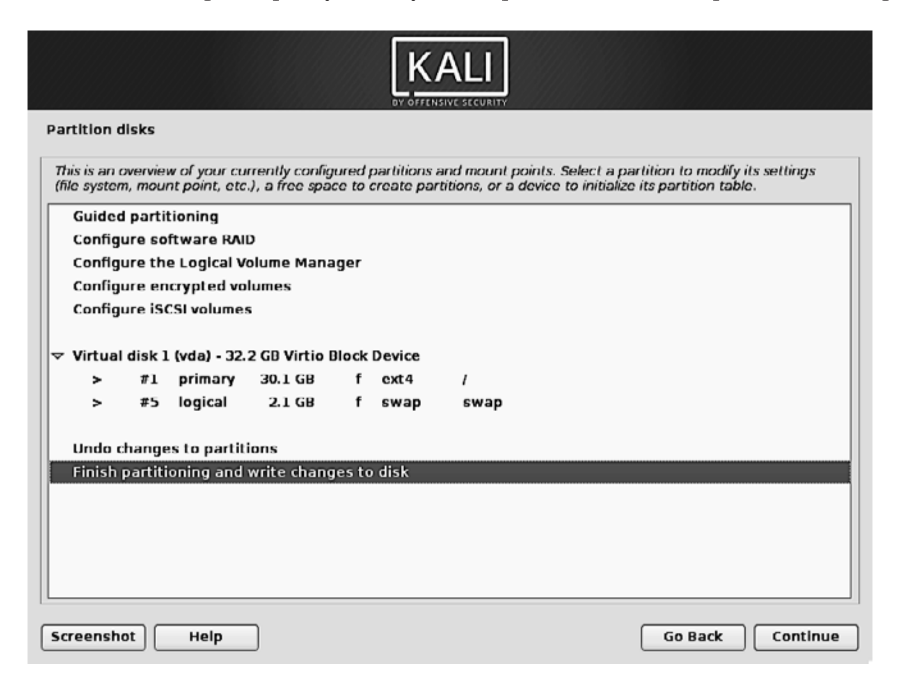


Рис. 4.11. Подтверждение разметки

Сжатие раздела Windows

Чтобы установить Kali Linux вместе с существующей операционной системой (Windows или другой), потребуется доступное неиспользуемое пространство на жестком диске для разделов, предназначенных для Kali. В большинстве случаев это означает сокращение существующего раздела и повторное применение освобожденного пространства.

Если вы задействуете ручной режим разметки, то установщик может легко сократить раздел Windows. Вам нужно только выбрать этот раздел и ввести его новый размер (это работает и с разделами FAT и NTFS).

Если вы менее опытный пользователь, работающий с системой с существующими данными, то будьте очень осторожны с этим методом настройки, так как легко совершать ошибки, которые могут привести к потере данных.

Первый экран в ручном установщике на самом деле такой же, как на рис. 4.11, за исключением того, что не содержит никаких новых разделов для создания. Вы можете добавить их по собственному усмотрению.

В первую очередь вы увидите опцию Guided partitioning (Управляемая разметка), а затем несколько вариантов конфигурации. Затем установщик покажет доступные диски, их разделы и все доступное свободное пространство, которое еще не было разделено. Как обычно, вы можете выбрать любой отображаемый элемент и нажать клавишу Enter, чтобы взаимодействовать с ним.

Если диск совершенно новый, то может потребоваться создать таблицу разделов. Для этого нужно выбрать диск. Затем вы увидите свободное место на диске.

Чтобы использовать это свободное пространство, вы должны выбрать его, и установщик предложит вам два способа создания разделов в данном пространстве (рис. 4.12).

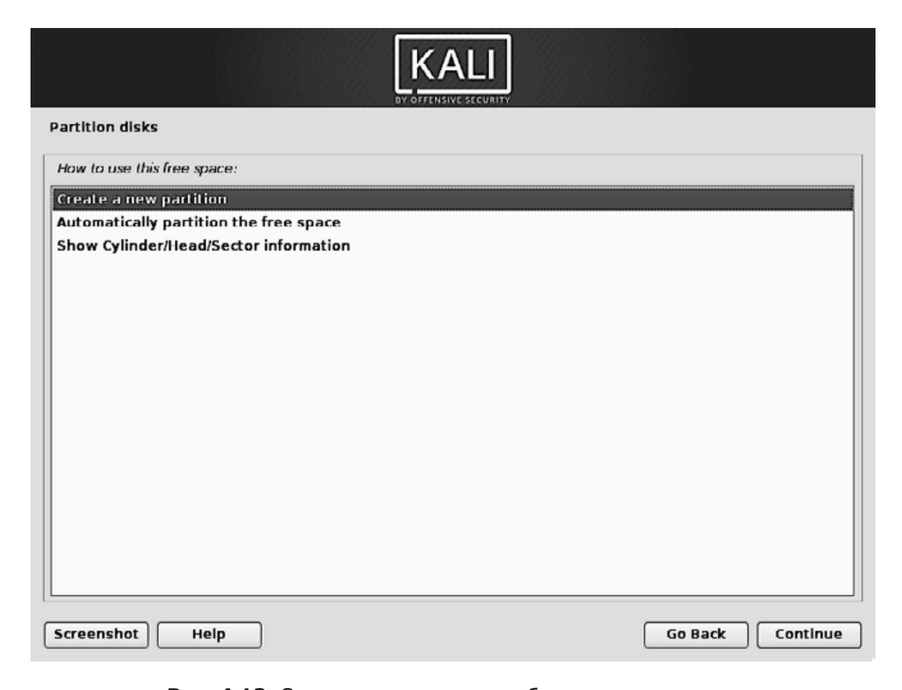


Рис. 4.12. Создание разделов в свободном пространстве

Первая опция создаст отдельный раздел с характеристиками (включая размер) по вашему выбору. Вторая использует все свободное пространство и создаст в нем несколько разделов с помощью мастера управляемой разметки (см. пункт «Управляемая разметка» выше). Данная опция особенно интересна, если вы хотите установить Kali вместе с другой операционной системой, но при этом не желаете управлять макетом разметки. В последней опции будут показаны номера цилиндра/головки/сектора начала и конца свободного пространства.

Выбрав пункт Create a new partition (Создать новый раздел), вы перейдете к сути ручной разметки. После выбора этой опции вам будет предложено указать размер раздела. Если на диске используется таблица разделов MS DOS, то вы получите возможность создать первичный или логический раздел. (Важно знать: вы можете

иметь только четыре первичных раздела, но множество логических. Раздел, содержащий /boot и, следовательно, ядро, должен быть первичным, логические разделы находятся в расширенном разделе, который использует один из четырех первичных разделов.) Затем вы увидите общий экран конфигурации раздела (рис. 4.13).

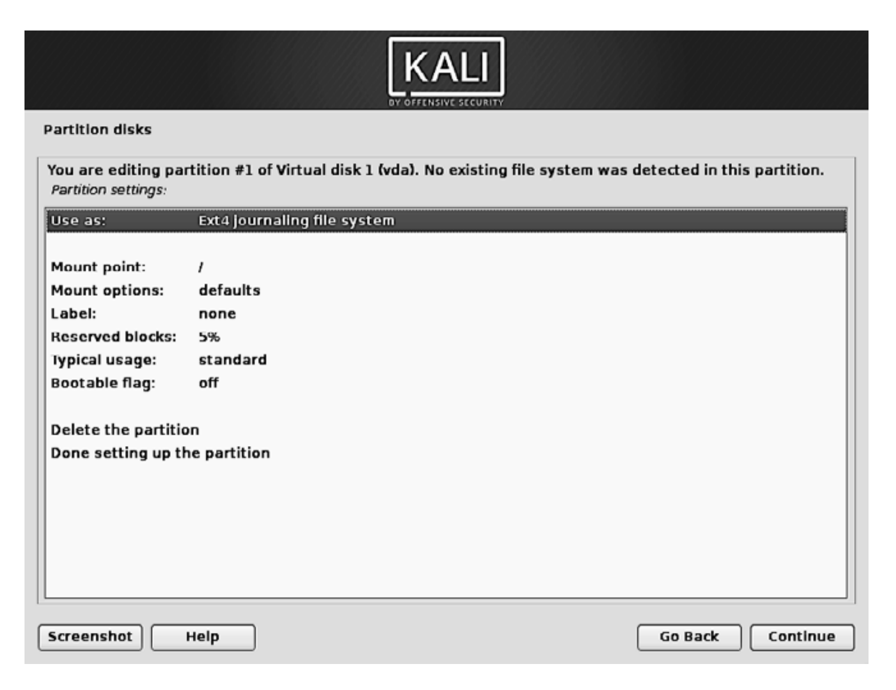


Рис. 4.13. Экран конфигурации раздела

Чтобы подытожить данный этап ручной разметки, посмотрим, какие действия вы можете выполнить с новым разделом.

- Отформатировать его и включить в дерево файлов, выбрав точку монтирования. Точка монтирования — это каталог, в котором будет размещаться содержимое файловой системы в выбранном разделе. Таким образом, раздел, смонтированный в /home/, традиционно предназначен для хранения пользовательских данных, а / известен как корень дерева файлов и, следовательно, корень раздела, на котором будет фактически установлена система Kali.
- Применять его как *раздел подкачки*. Когда в ядре Linux недостаточно свободной памяти, оно будет хранить неактивные части ОЗУ в специальном разделе подкачки на жестком диске. Подсистема виртуальной памяти делает это прозрачным для приложений. Чтобы симулировать дополнительную память, Windows использует файл подкачки, который содержится непосредственно в файловой системе. Напротив, Linux задействует раздел, предназначенный для этой цели, отсюда и термин «раздел подкачки».

- Сделать его «физическим томом для шифрования», чтобы защитить конфиденциальность данных на определенных разделах. Этот случай автоматизирован в управляемой разметке. Дополнительную информацию см. ниже в подразделе «Установка на полностью зашифрованную файловую систему» текущего раздела.
- Сделать его «физическим томом для LVM» (не описано в этой книге). Обратите внимание, что эта функция используется управляемой разметкой при настройке зашифрованных разделов.
- Задействовать его как RAID-устройство (не описано в данной книге).
- Не использовать раздел и оставить его неизменным.

По завершении вы можете либо отказаться от ручной разметки, выбрав пункт Undo changes to Partitions (Отменить изменения в разделах), или записать изменения на диск, выбрав пункт Finish partitioning and write changes to disk (Завершить разметку и записать изменения на диск) на экране ручного установщика (см. рис. 4.11).

Копирование live-образа

Этот шаг, не требующий какого-либо взаимодействия с пользователем, копирует содержимое live-образа в целевую файловую систему (рис. 4.14).

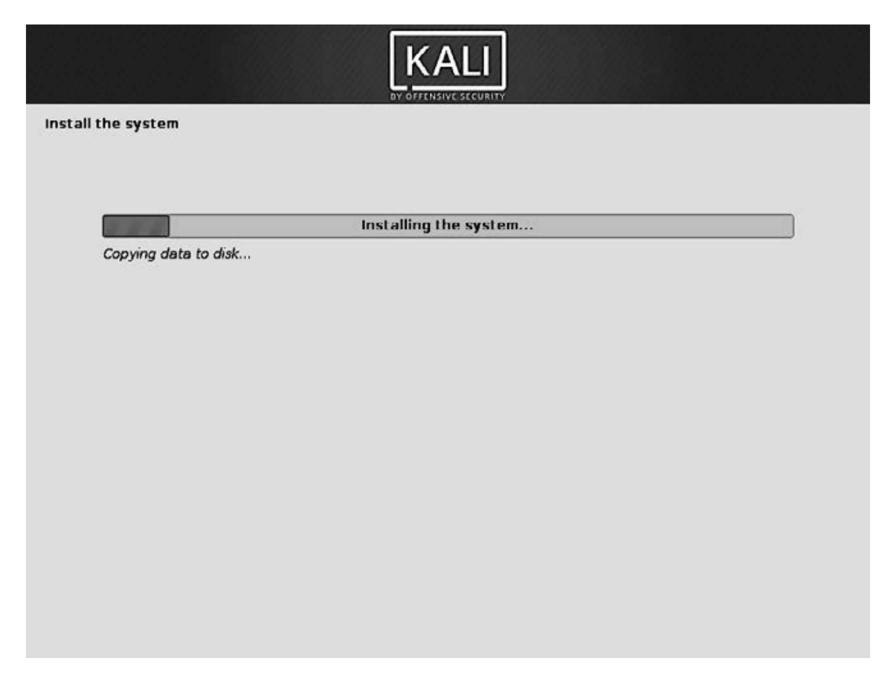


Рис. 4.14. Копирование данных из live-образа

Настройка менеджера пакетов (apt)

Чтобы иметь возможность устанавливать дополнительное программное обеспечение, необходимо настроить APT и указать, где находятся пакеты Debian. В Kali этот шаг в основном неинтерактивный, поскольку мы размещаем зеркало на https://www.kali.org/. Вам просто нужно подтвердить, хотите ли вы использовать данное зеркало (рис. 4.15). Не применяя его, вы не сможете установить дополнительные пакеты с помощью `apt`, если позднее не настроите репозиторий пакетов.

Рис. 4.15. Использовать сетевое зеркало?

Если вы хотите задействовать локальное зеркало вместо https://www.kali.org/, то можете указать его имя в командной строке ядра (при загрузке) с синтаксисом, подобным этому: `mirror/http/hostname=`*`мое.собственное.зеркало`*.

Наконец, программа предлагает использовать *HTTP-прокси*, как показано на рис. 4.16. Прокси-сервер HTTP — сервер, который перенаправляет HTTP-запросы для пользователей сети. Иногда это помогает ускорить скачивание, сохранив копию файлов, которые были перенесены через него (мы говорим о кэшировании прокси). В ряде случаев это единственный способ получить доступ к внешнему веб-серверу; в таких ситуациях установщик сможет скачивать пакеты Debian, только если вы правильно заполните данное поле во время установки. Неуказанный адрес прокси-сервера приведет к тому, что установщик попытается подключиться непосредственно к Интернету.



Рис. 4.16. Использовать HTTP-прокси

Затем автоматически будут скачаны файлы `Packages.xz` и `Sources.xz`, что позволит обновить список пакетов, распознанных APT.

Установка загрузчика GRUB

Системный загрузчик — это первая программа, запущенная BIOS. Данная программа загружает ядро Linux в память и затем запускает его. Системный загрузчик часто предлагает меню, которое позволяет выбрать загружаемое ядро или операционную систему.

Благодаря своему техническому превосходству GRUB является загрузчиком по умолчанию, установленным Debian: работает с большинством файловых систем и поэтому не требует обновления после каждой установки нового ядра, поскольку считывает его конфигурацию во время загрузки и находит его точную позицию.

Вы должны установить GRUB в Master Boot Record (MBR), если у вас еще не установлена другая система Linux, которая знает, как загружать Kali Linux. На рис. 4.17 показано: изменение MBR приведет к тому, что нераспознанные операционные системы, зависящие от него, будут незагружаемыми, пока вы не исправите конфигурацию GRUB.

На данном этапе (рис. 4.18) вы должны выбрать, какое устройство GRUB будет установлено. Таковым должен быть ваш текущий загрузочный диск.

KALI
BY OFFENSIVE SECURITY

Install the GRUB boot loader on a hard disk

It seems that this new installation is the only operating system on this computer. If so, it should be safe to install the GRUB boot loader to the master boot record of your first hard drive.

Warning: If the installer failed to detect another operating system that is present on your computer, modifying the master boot record will make that operating system temporarily unbootable, though GRUB can be manually configured later to boot it.

Install the GRUB boot loader to the master boot record?

No

Yes

Screenshot | Go Back | Continue

Рис. 4.17. Установить загрузчик GRUB на жесткий диск

KALI
BY OFFENSIVE SECURITY

Install the GRUB boot loader on a hard disk

You need to make the newly installed system bootable, by installing the GRUB boot loader on a bootable device. The usual way to do this is to install GRUB on the master boot record of your first hard drive. If you prefer, you can install GRUB elsewhere on the drive, or to another drive, or even to a floppy.

Device for boot loader installation:

Enter device manually

/dev/vda

Screenshot | Go Back | Continue

Рис. 4.18. Устройство для установки системного загрузчика

По умолчанию в меню загрузки, предлагаемом GRUB, отображаются все установленные ядра Linux, а также любые другие обнаруженные операционные системы. Поэтому вы должны принять предложение об установке ядра в MBR. Хранение старых версий ядра сохраняет возможность загрузки системы, если последнее установленное ядро повреждено или плохо адаптировано к оборудованию. Так что мы рекомендуем сохранить несколько старых версий ядра.

Внимание: системный загрузчик и двойная загрузка

Данная фаза процесса установки обнаруживает операционные системы, которые уже установлены на компьютере, и автоматически добавляет соответствующие пункты в меню загрузки. Однако не все программы установки делают это.

В частности, если после этого вы установите (или переустановите) Windows, то загрузчик подвергнется удалению. Kali сохранит свое расположение на жестком диске, но больше не будет доступен из меню загрузки. Вам придется запустить установщик Kali с параметром rescue/enable = true в командной строке ядра, чтобы переустановить загрузчик. Эта операция подробно описана в руководстве по установке Debian (https://www.debian.org/releases/stable/amd64/ch08s07.html).

Завершение установки и перезагрузка

Теперь, когда установка завершена, программа попросит вас извлечь DVD из дисковода (или отсоединить USB-накопитель), чтобы ваш компьютер смог загрузить новую систему Kali после того, как программа установки перезапустит систему (рис. 4.19).

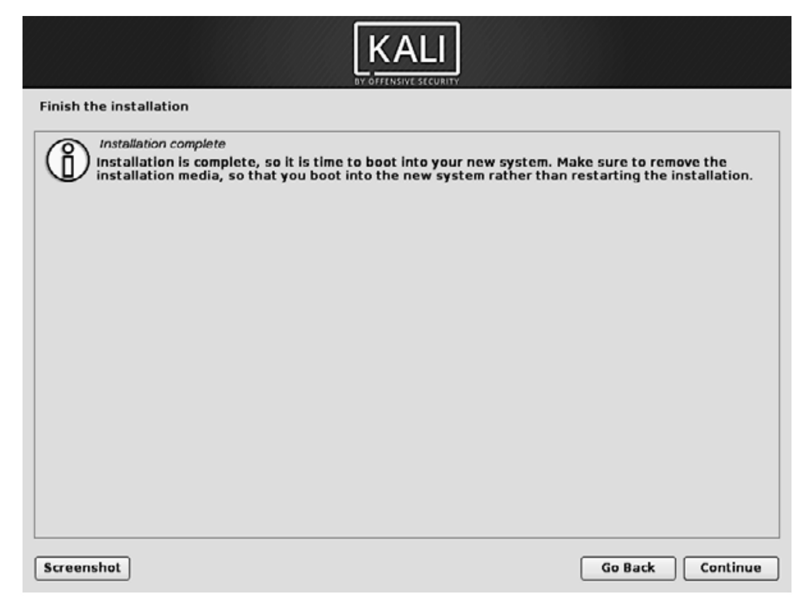


Рис. 4.19. Установка завершена

Наконец установщик выполнит некую очистку, например удалит пакеты, специфичные для создания live-среды.

Установка на полностью зашифрованную файловую систему

Чтобы гарантировать конфиденциальность вашей информации, вы можете настроить зашифрованные разделы. Это защитит данные, если ваш ноутбук или жесткий диск будут потеряны или украдены. Инструмент разбиения на разделы (как управляемый, так и ручной) может помочь.

Режим управляемого разделения сочетает использование двух технологий: Linux Unified Key Setup (LUKS) для шифрования разделов и Logical Volume Management (управление логическими томами — LVM) для динамического управления хранилищем. Обе функции также можно настроить в режиме разделения вручную.

Введение в LVM

Сначала обсудим LVM. Используя терминологию LVM, *виртуальный раздел* представляет собой логический том, который выступает в качестве части *группы томов* или объединения нескольких физических томов. Последние являются реальными разделами (или виртуальными разделами, экспортируемыми другими абстракциями, такими как программное RAID-устройство или зашифрованный раздел).

Благодаря отсутствию различий между «физическими» и «логическими» разделами LVM позволяет создавать «виртуальные» разделы, которые охватывают несколько дисков. Выгода двоякая: размер разделов больше ограничен не отдельными дисками, а их совокупным объемом, и можно изменить размер существующих разделов в любое время, например после добавления дополнительного диска.

Этот метод работает очень просто: каждый том, будь то физический или логический, разбивается на блоки равного размера, которые коррелирует LVM. Добавление нового диска приведет к созданию нового физического тома, обеспечивающего новые блоки, которые могут быть связаны с любой группой томов. Всем разделам в группе томов разрешается в полной мере использовать дополнительное выделенное пространство.

Введение в LUKS

Чтобы защитить ваши данные, можете добавить уровень шифрования под своей файловой системой. Linux (и в частности драйвер `dm-crypt`) использует устройство `mapper` для создания виртуального раздела (имеющего защищенное содержимое) на основе базового раздела, который будет хранить данные в зашифрованном виде (благодаря LUKS). LUKS стандартизирует хранение зашифрованных данных, а также метаинформацию, которая указывает на применяемые алгоритмы шифрования.

Зашифрованный раздел подкачки	Когда используется зашифрованный раздел, ключ шифрования сохраняется в памяти (ОЗУ), а в спящем режиме ноутбук копирует ключ вместе с прочим содержимым ОЗУ в раздел подкачки жесткого диска. Поскольку любой, кто имеет доступ к файлу подкачки (включая техника или вора), может извлечь ключ и расшифровать ваши данные, то файл подкачки должен быть защищен с помощью шифрования. По этой причине установщик выдаст предупреждение, если вы попытаетесь использовать зашифрованный раздел вместе с незашифрованным разделом подкачки.

Настройка зашифрованных разделов

Процесс установки для зашифрованного LVM выглядит так же, как стандартная установка, за исключением этапа разбиения на разделы (рис. 4.20), где теперь нужно выбрать пункт Guided — use entire disk and set up encrypted LVM (Управляемый — использовать весь диск и настроить зашифрованный LVM). Конечным результатом станет система, которую нельзя загрузить или получить к ней доступ до тех пор, пока не будет предоставлен пароль шифрования. Это зашифрует и защитит данные на вашем диске.

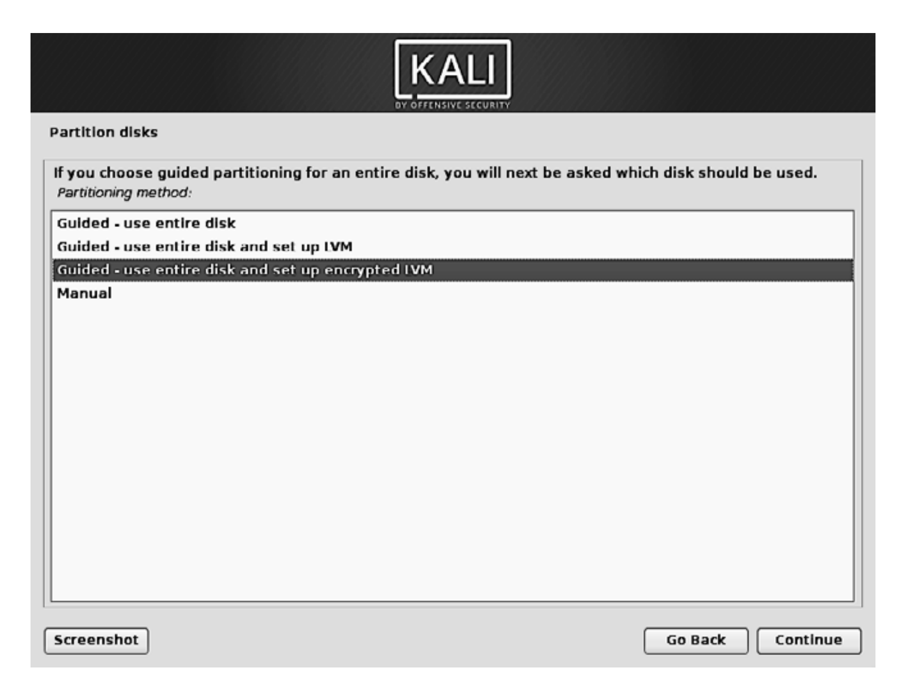


Рис. 4.20. Управляемое разделение с зашифрованным LVM

Установщик управляемого разделения автоматически назначит физический раздел для хранения зашифрованных данных (рис. 4.21). На данном этапе установщик подтвердит изменения до того, как они будут записаны на диск.

KALI
BY OFFENSIVE SECURITY

Partition disks

Before the Logical Volume Manager can be configured, the current partitioning scheme has to be written to disk. These changes cannot be undone.

After the Logical Volume Manager is configured, no additional changes to the partitioning scheme of disks containing physical volumes are allowed during the installation. Please decide if you are satisfied with the current partitioning scheme before continuing.

The partition tables of the following devices are changed:
Virtual disk 1 (vda)

Write the changes to disks and configure LVM?

No

Yes

Screenshot

Continue

Рис. 4.21. Подтвердить изменения в таблице разделов

Затем новый раздел инициализируется случайными данными (рис. 4.22). Это делает области, которые содержат данные, неотличимыми от неиспользуемых областей, что затрудняет обнаружение и последующую атаку зашифрованных данных.

Затем установщик попросит ввести пароль шифрования (рис. 4.23). Чтобы просмотреть содержимое зашифрованного раздела, вам придется вводить этот пароль при каждой перезагрузке системы. Обратите внимание на предупреждение в установщике: ваша зашифрованная система будет столь же сильна, как и данный пароль.

Теперь в инструменте разбиения есть доступ к новому виртуальному разделу, содержимое которого зашифровано в базовом физическом разделе. Поскольку LVM использует этот новый раздел в качестве физического тома, то может защищать несколько разделов (или LVM-логических томов) с одним и тем же ключом шифрования, в том числе раздел подкачки (см. выше врезку «Зашифрованный раздел подкачки»). Здесь LVM применяется не с целью упростить расширение размера хранилища, а только для удобства косвенной адресации, позволяющей разделить один зашифрованный раздел на несколько логических томов.

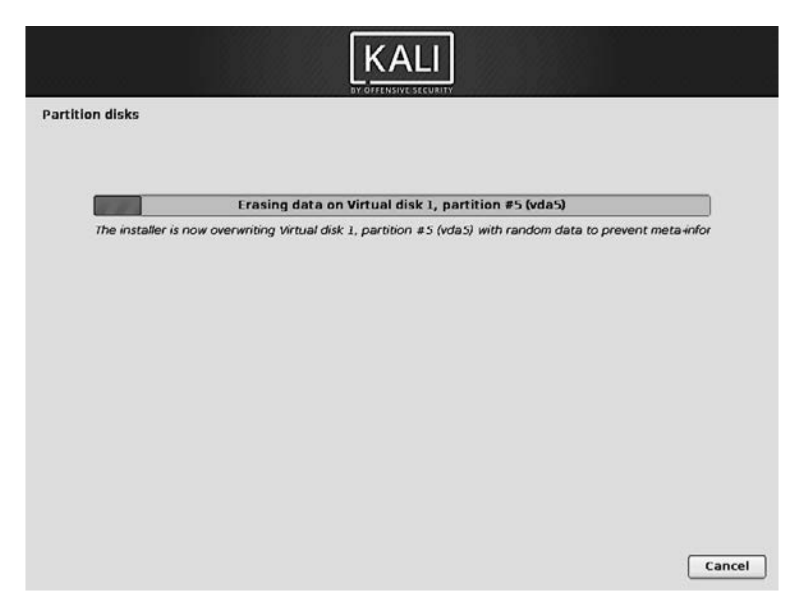


Рис. 4.22. Удаление данных на зашифрованном разделе

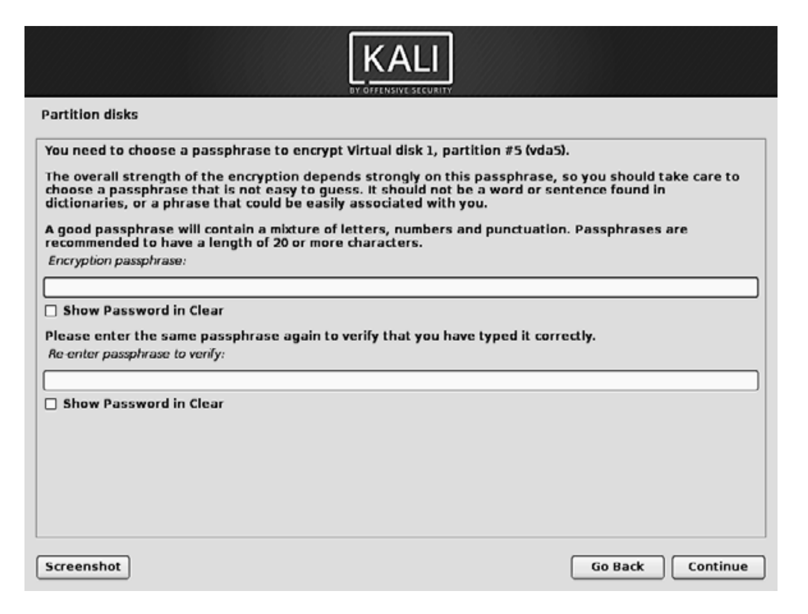


Рис. 4.23. Введите пароль шифрования

Окончание управляемого разбиения с зашифрованным LVM

На следующем шаге будет показана получившаяся схема разбиения (рис. 4.24), чтобы вы могли изменить настройки в соответствии с потребностями.

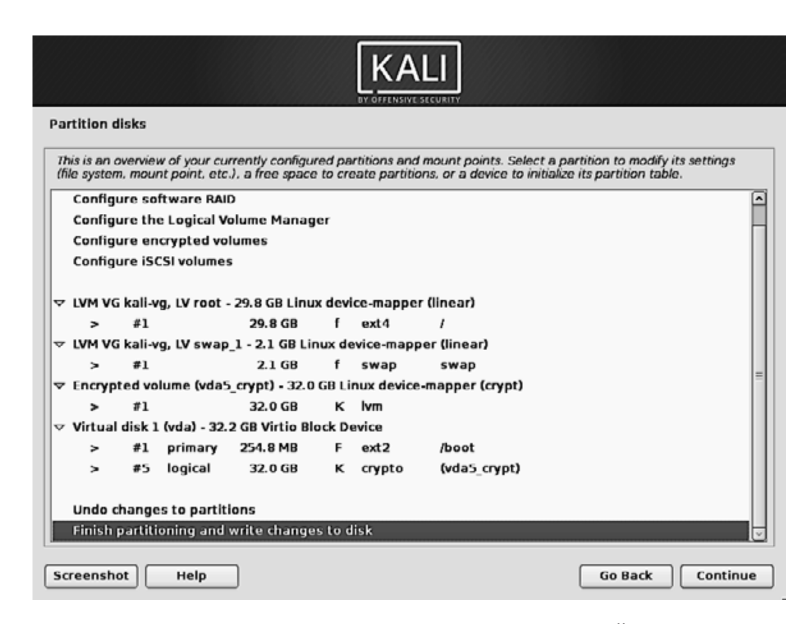


Рис. 4.24. Подтверждение разбиения для зашифрованной установки LVM

Наконец, после проверки настроек раздела инструмент просит подтвердить внесенные на диск изменения (рис. 4.25).

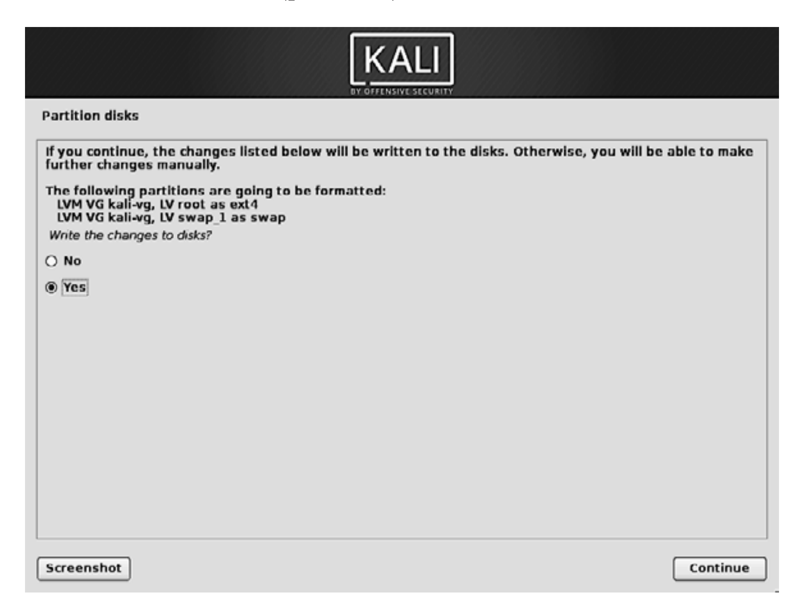


Рис. 4.25. Подтверждение форматирования разделов

Процесс установки продолжается как обычно, что описано в пункте «Настройка менеджера пакетов (apt)» ранее в этой главе.

4.3. Автоматическая установка

Установщики Debian и Kali очень модульные: на базовом уровне они просто выполняют множество сценариев (упакованных в крошечные пакеты, называемые udeb (μdeb или микро-deb)) один за другим. Каждый сценарий использует `debconf` (см. врезку «Инструмент debconf» в главе 8), который взаимодействует с вами, пользователем, и сохраняет параметры установки. Благодаря этому установщика также можно автоматизировать с помощью `debconf preseeding` — функции, которая позволяет давать автоматические ответы на вопросы установки.

Автоматические ответы

Существует несколько способов предоставить ответы установщику. Каждый метод имеет свои преимущества и недостатки. В зависимости от момента, когда происходит автоматизация, вопросы для предварительных ответов различаются.

С помощью параметров загрузки

Вы можете указать предварительный ответ на любой вопрос установщика, используя параметры загрузки, которые попадают в командную строку ядра, с доступом через `/proc/cmdline`. Отдельные загрузчики позволят редактировать эти параметры в интерактивном режиме (что полезно для целей тестирования), но если хотите, чтобы изменения носили постоянный характер, то придется изменить настройки загрузчика.

Вы можете напрямую использовать полный идентификатор вопросов `debconf` (например, `debian-installer/language=en`) или аббревиатуры для наиболее распространенных вопросов (допустим, `language=en` или `hostname=duke`). Полный список псевдонимов см. в руководстве по установке Debian (https://www.debian.org/releases/stable/amd64/apbs02#preseed-aliases).

Нет ограничений на вопросы, ответы для которых можно указать предварительно, поскольку параметры загрузки доступны с самого начала процесса установки и обрабатываются очень рано. Однако количество загрузочных параметров ограничено до 32-х, и некоторые из них уже используются по умолчанию. Кроме того, важно понимать, что изменение конфигурации загрузчика иногда бывает нетривиальным.

В разделе 9.3 вы также узнаете, как изменить конфигурацию Isolinux при создании собственного ISO-образа Kali.

С помощью файла пресидинга в Initrd

Вы можете добавить файл с именем `preseed.cfg` в корень `initrd` установщика (это `initrd`, который используется для запуска установщика). Обычно данное действие требует восстановления исходного пакета `debian-installer` для генерации новых версий `initrd`. Тем не менее `live-build` предлагает сделать это удобным способом, подробно описанным в разделе 9.3.

Данный метод тоже не имеет никаких ограничений на вопросы, ответы для которых вы можете предварительно указать, поскольку файл пресидинга доступен сразу после загрузки. В Kali мы уже используем эту функцию для настройки поведения официального установщика Debian.

С помощью файла пресидинга на загрузочном носителе

Вы можете добавить файл пресидинга на загрузочный носитель (CD или USB-накопитель); загрузка предварительных ответов на вопросы произойдет, как только будет установлен носитель, а значит, сразу после вопросов о языке и раскладке клавиатуры. Параметр загрузки `preseed/file` может использоваться для указания местоположения файла пресидинга (например, `/cdrom/preseed.cfg` при установке с компакт-диска или `/hd-media/preseed.cfg` при установке с USB-ключа).

Нельзя предварительно задать ответы для параметров языка и страны, поскольку файл пресидинга загружается позже, после загрузки драйверов оборудования. Что касается позитивной стороны, то `live-build` упрощает размещение дополнительного файла в сгенерированных образах ISO (см. раздел 9.3).

С помощью файла пресидинга, загруженного из сети

Вы можете обеспечить доступ к файлу пресидинга в сети через веб-сервер и сообщить установщику о скачивании этого файла, добавив параметр загрузки `preseed/url=http://`*`сервер`*`/preseed.cfg` (или с помощью псевдонима `url`).

Однако при использовании данного метода помните, что сначала необходимо настроить сеть. Это значит, что связанные с сетью вопросы `debconf` (в частности, имя хоста и имя домена) и все предыдущие вопросы (например, язык и страна) не могут быть запрограммированы с помощью данного метода. Он чаще всего применяется в сочетании с параметрами загрузки, определяющими ответы на эти конкретные вопросы.

Этот метод пресидинга — наиболее гибкий, так как вы можете изменить конфигурацию установки, не затрагивая установочный носитель.

Задержка вопросов о языке, стране, клавиатуре

Чтобы преодолеть ограничение, из-за которого нельзя предварительно отвечать на вопросы о языке, стране и клавиатуре, вы можете добавить параметр загрузки auto-install/enable=true (или auto=true). С помощью этого параметра вопросы будут заданы позже, после настройки сети и, следовательно, после скачивания файла пресидинга.

Недостатком данного параметра является то, что первые шаги (в частности, конфигурация сети) всегда будут происходить на английском языке, и при наличии ошибок пользователю придется работать с экраном на этом языке (с клавиатурой в режиме QWERTY).

Создание файла пресидинга

Файл пресидинга — это обычный текстовый файл, в котором каждая строка содержит ответ на один вопрос Debconf. Строка разделяется на четыре поля, разграниченные пробельными символами (пробелами или отступами). Рассмотрим в качестве образца строку `d-i mirror/suite string kali-rolling`.

- В первом поле указывается владелец вопроса. Например, `d-i` используется для вопросов, относящихся к установщику. Вы также можете увидеть имя пакета для вопросов, исходящих из пакетов Debian (как в этом примере: `atftpd atftpd/use_inetd boolean false`).
- Второе поле — идентификатор вопроса.
- В третьем поле указан тип вопроса.
- Четвертое, заключительное, поле содержит значение ожидаемого ответа. Обратите внимание: оно должно быть отделено от третьего поля одним пробелом; дополнительные пробельные символы считаются частью значения.

Самый простой способ написать файл пресидинга — установить систему вручную. Затем команда `debconf-get-selections --installer` выдаст ответы, которые вы предоставили установщику. Вы можете получить ответы и относительно других пакетов с помощью `debconf-get-selections`. Тем не менее более чистым решением является запись файла пресидинга вручную, начиная с примера, а затем задействуя документацию. При таком подходе могут быть предустановлены ответы только на те вопросы, по которым нужно отменить ответ по умолчанию. Укажите загрузочный параметр `priority=critical`, чтобы дать `Debconf` команду задавать только критические вопросы и использовать ответ по умолчанию для других.

Руководство по установке

В руководстве по установке Debian, доступном в Интернете, содержится подробная документация по использованию файла пресидинга в приложении (https://www.debian.org/releases/stable/amd64/apb.html). Оно также содержит подробный и прокомментированный образец файла, который может служить базой для локальных настроек (https://www.debian.org/releases/stable/example-preseed.txt).

Однако обратите внимание: приведенные выше ссылки документируют стабильную версию Debian и Kali использует тестовую версию, поэтому вы можете столкнуться с небольшими различиями. На сайте проекта Debian-installer (http://d-.alioth.debian.org/manual/en.amd64/apb.html) размещено руководство по установке. Оно может быть более актуальным.

4.4. Установка на ARM-устройства

Kali Linux работает на самых разных устройствах на базе ARM (например, ноутбуках, встроенных компьютерах и платах разработчиков), но вы не можете использовать традиционный установщик Kali на этих устройствах, поскольку они

часто имеют особые требования в отношении конфигурации ядра или системного загрузчика.

Чтобы сделать эти устройства более доступными для пользователей Kali, Offensive Security разработала сценарии (https://github.com/offensive-security/kali-arm-build-scripts) для создания образов дисков, которые готовы к применению с различными ARM-устройствами. Эти образы представлены для скачивания на сайте https://www.offensive-security.com/kali-linux-arm-images/.

Доступность указанных образов значительно упрощает задачу установки Kali на ARM-устройстве. Ниже перечислены основные шаги.

1. Скачайте образ для своего ARM-устройства и убедитесь, что контрольная сумма соответствует той, которая указана на сайте (см. подраздел «Проверка целостности и подлинности» раздела 2.1). Обратите внимание, что образы обычно `xz`-сжаты; обязательно распакуйте их с помощью утилиты `unxz`.
2. В зависимости от слота расширения хранилища, доступного на вашем конкретном ARM-устройстве, приобретите SD-карту, micro-SD-карту или встроенный модуль мультимедийного контроллера (eMMC) емкостью не менее 8 Гбайт.
3. Скопируйте скачанный образ на устройство хранения с помощью команды `dd`. Это похоже на процесс копирования ISO-образа на USB-накопитель (см. подраздел «Копирование образа на DVD- или USB-накопитель» раздела 2.1).
4. Подключите SD- или eMMC-карту к ARM-устройству.
5. Загрузите ARM-устройство и войдите в него (пользователь `root`, пароль `toor`). При отсутствии подключенного экрана нужно будет определить IP-адрес, назначенный через DHCP, и подключиться к этому адресу через SSH. На некоторых серверах DHCP есть инструменты или веб-интерфейсы, указывающие текущую аренду. Если у вас нет ничего подобного, примените сниффер для поиска аренды трафика DHCP.
6. Измените пароль `root`-пользователя и сгенерируйте новые ключи хоста SSH, особенно если устройство будет постоянно работать в общедоступной сети! Шаги относительно просты, см. врезку «Создание новых хост-ключей SSH» в главе 5.
7. Наслаждайтесь ARM-устройством, работающим под управлением операционной системы Kali Linux!

Особые случаи и подробная документация

Эти инструкции носят общий характер и актуальны для большинства устройств, однако всегда есть исключения. Например, для Chromebook требуется режим разработчика, а для других устройств — специальное нажатие клавиши для загрузки с внешнего носителя.

Поскольку новые ARM-устройства появляются относительно часто и их характеристики весьма динамичны, то мы не будем описывать здесь конкретные инструкции по установке для различных ARM-устройств. Вместо этого обратитесь к разделу Kali on ARM на сайте документации Kali для получения информации о каждом ARM-устройстве, поддерживаемом Offensive Security (https://docs.kali.org/category/kali-on-arm).

4.5. Устранение неполадок установки

Установщик достаточно надежный, но вы можете столкнуться с неполадками или внешними проблемами, такими как проблемы с сетью, плохие зеркала и нехватка дискового пространства. Поэтому будет весьма полезно уметь устранять проблемы, возникающие в процессе установки.

Когда в программе установки происходит сбой, она покажет вам довольно бесполезный экран, такой как на рис. 4.26.

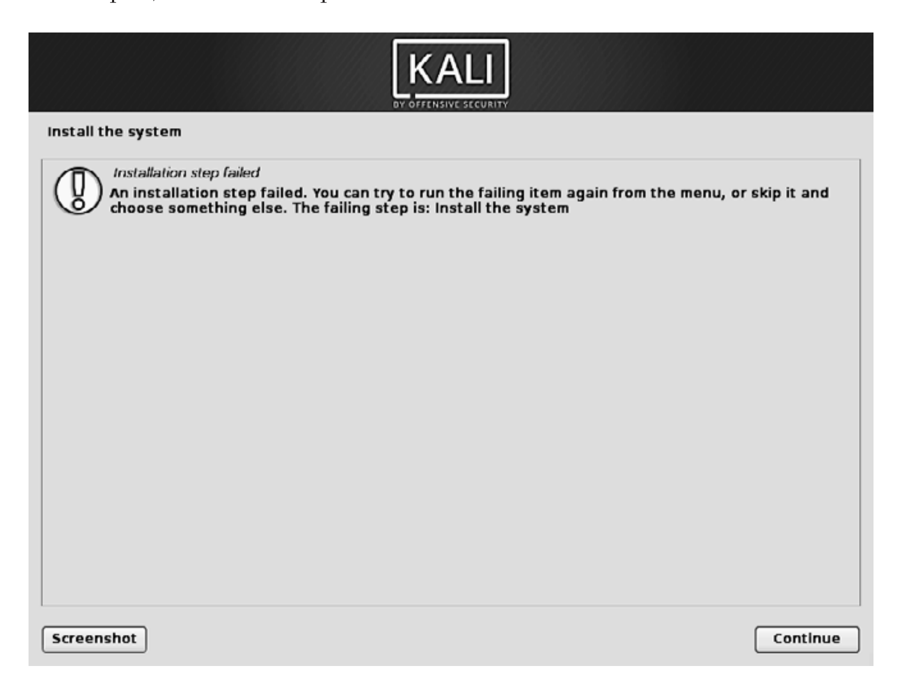


Рис. 4.26. Шаг установки не выполнен

На данном этапе хорошо знать, что установщик использует несколько виртуальных консолей: главный экран, который вы видите, запускается либо на пятой консоли (для графического установщика, Ctrl+Shift+F5), либо на первой (для текстового установщика, Ctrl+Shift+F1). В обоих случаях четвертая консоль (Ctrl+Shift+F4) отображает записи того, что происходит, и обычно вы можете увидеть там более полезное сообщение об ошибке, такое как на рис. 4.27, показывающем что у установщика закончилось дисковое пространство.

Вторая и третья консоли (Ctrl+Shift+F2 и Ctrl+Shift+F3 соответственно) содержат оболочки, пригодные для более детального изучения текущей ситуации. Большинство инструментов командной строки предоставляется BusyBox, поэтому набор функций довольно ограничен, но достаточен, чтобы выяснить большинство проблем, с которыми вы, вероятно, столкнетесь.

```
tion:
Apr 15 19:04:24 main-menu[833]: (process:5559): line 88:
Apr 15 19:04:24 main-menu[833]: (process:5559): /lib/partman/active_partition/copy/choices: not foun
d
Apr 15 19:04:24 main-menu[833]: (process:5559):
Apr 15 19:04:24 main-menu[833]: (process:5559): /lib/partman/choose_partition/60partition_tree/do_op
tion:
Apr 15 19:04:24 main-menu[833]: (process:5559): line 88:
Apr 15 19:04:24 main-menu[833]: (process:5559): /lib/partman/active_partition/copy/choices: not foun
d
Apr 15 19:04:24 main-menu[833]: (process:5559):
Apr 15 19:04:24 main-menu[833]: (process:5559): /lib/partman/free_space/50new/do_option:
Apr 15 19:04:24 main-menu[833]: (process:5559): line 226:
Apr 15 19:04:24 main-menu[833]: (process:5559): /lib/partman/active_partition/copy/choices: not foun
d
Apr 15 19:04:24 main-menu[833]: (process:5559):
Apr 15 19:04:24 main-menu[833]: (process:5559): /lib/partman/free_space/50new/do_option:
Apr 15 19:04:24 main-menu[833]: (process:5559): line 226:
Apr 15 19:04:24 main-menu[833]: (process:5559): /lib/partman/active_partition/copy/choices: not foun
d
Apr 15 19:04:24 main-menu[833]: (process:5559):
Apr 15 19:04:24 main-menu[833]: DEBUG: resolver (libgcc1): package doesn't exist (ignored)
Apr 15 19:04:24 main-menu[833]: INFO: Menu item 'live-installer' selected
Apr 15 19:04:24 base-installer: info: Using squashfs support for /cdrom/live/filesystem.squashfs
Apr 15 19:04:24 anna-install: Installing squashfs-modules
Apr 15 19:04:24 anna[8545]: DEBUG: resolver (kernel-image-4.3.0-kali1-amd64-di): package doesn't exi
st (ignored)
Apr 15 19:04:24 anna[8545]: DEBUG: retrieving squashfs-modules-4.3.0-kali1-amd64-di 4.3.3-5kali4
Apr 15 19:04:24 kernel: [  165.758382] squashfs: version 4.0 (2009/01/31) Phillip Lougher
Apr 15 19:04:24 kernel: [  165.764051] loop: module loaded
Apr 15 19:04:45 base-installer: error: The tar process copying the live system failed (only 9238 out
 of 119223 files have been copied, last file was ).
Apr 15 19:04:45 main-menu[833]: (process:8491): tar: write error: No space left on device
Apr 15 19:04:45 main-menu[833]: (process:8491): tar: write error: Broken pipe
Apr 15 19:04:45 main-menu[833]: WARNING **: Configuring 'live-installer' failed with error code 1
Apr 15 19:04:45 main-menu[833]: WARNING **: Menu item 'live-installer' failed.
_
```

Рис. 4.27. Экран журнала установщика

Что можно сделать в оболочке установщика

Вы можете проверить и изменить базу данных debconf с помощью команд debconf-get и debconf-set. Они особенно удобны для тестирования пресидинговых значений.

Вы можете проверить любой файл (например, полный журнал установки, доступный в /var/log/syslog) через команду cat или more; редактировать любой файл с помощью редактора nano, включая все файлы, установленные в систему. Корневая файловая система будет смонтирована/установлена после завершения этапа разметки процесса установки.

После настройки сетевого доступа вы можете использовать команды wget и nc (netcat — сетевой каталог) для извлечения и экспорта данных по сети.

Нажав кнопку Continue (Продолжить) на главном экране сбоя установщика (см. рис. 4.26), вы возвратитесь на экран, который в обычной ситуации никогда не увидите (главное меню, показанное на рис. 4.28), позволяющий запускать установку шаг за шагом. Если вам удалось устранить проблему через доступ к оболочке (поздравляем!), то можете повторить неудавшийся шаг.

Если попытки решить проблему не приносят нужного результата, то может потребоваться файл с сообщением об ошибке. Отчет должен содержать и журналы установщика, которые можно получить с помощью команды Save debug logs (Сохранить журналы отладки) главного меню. Она предлагает несколько способов экспорта журналов, как показано на рис. 4.29.

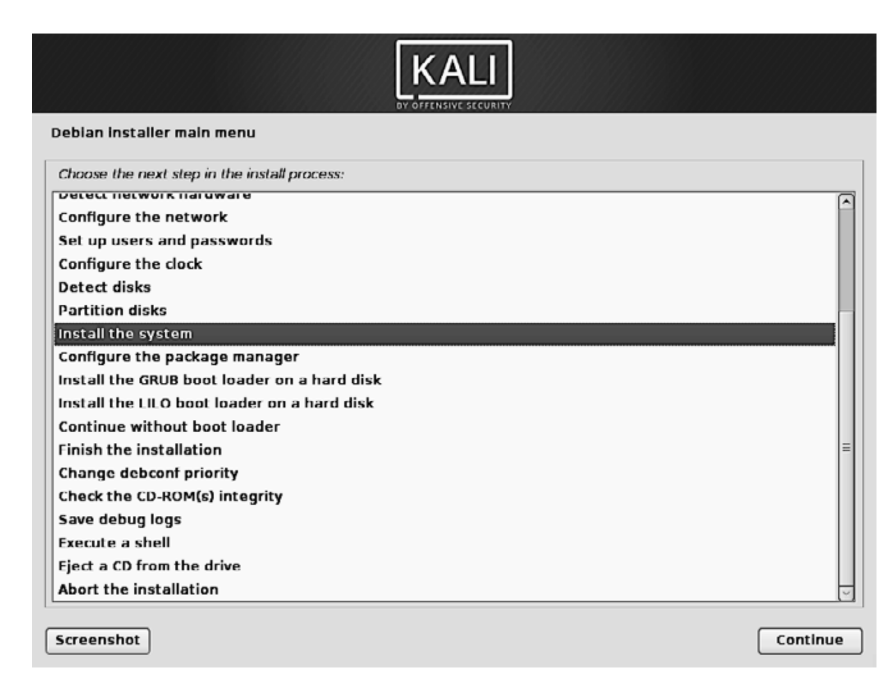


Рис. 4.28. Главное меню установщика

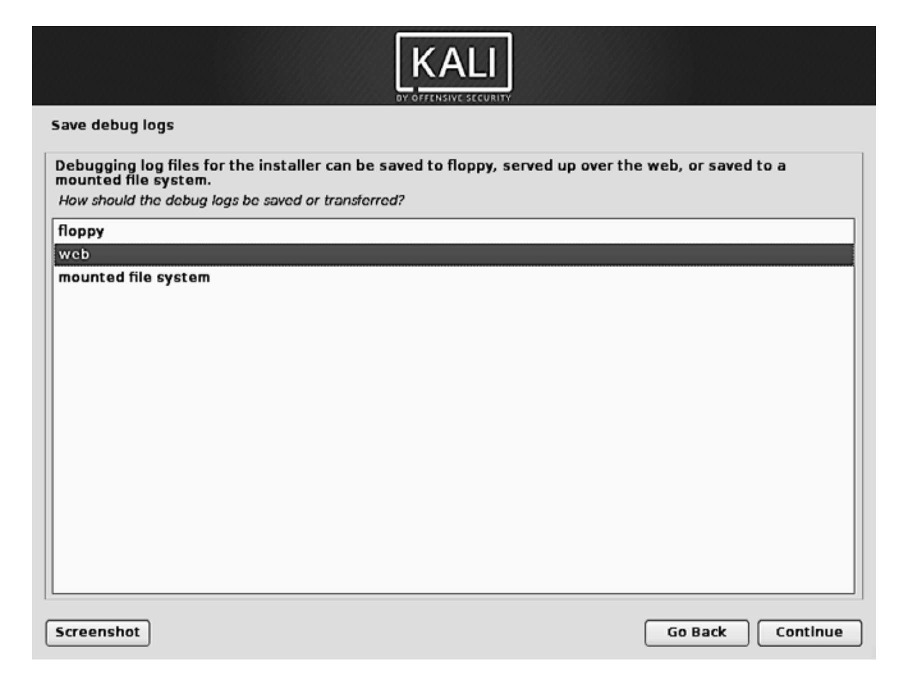


Рис. 4.29. Сохранить журналы отладки (1/2)

Самый удобный способ (его мы и рекомендуем) — позволить установщику запустить веб-сервер, на котором размещаются файлы журнала (рис. 4.30). Затем вы можете запустить браузер с другого компьютера в той же сети и скачать все файлы журналов и снимки экрана, сделанные вами с помощью кнопки Screenshot (Снимок экрана), доступной на каждом экране.

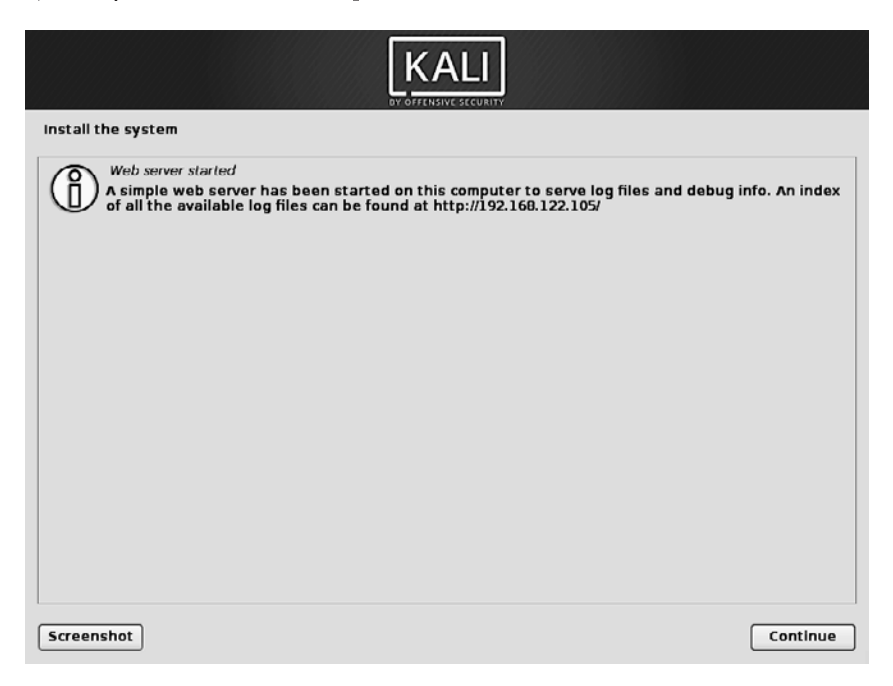
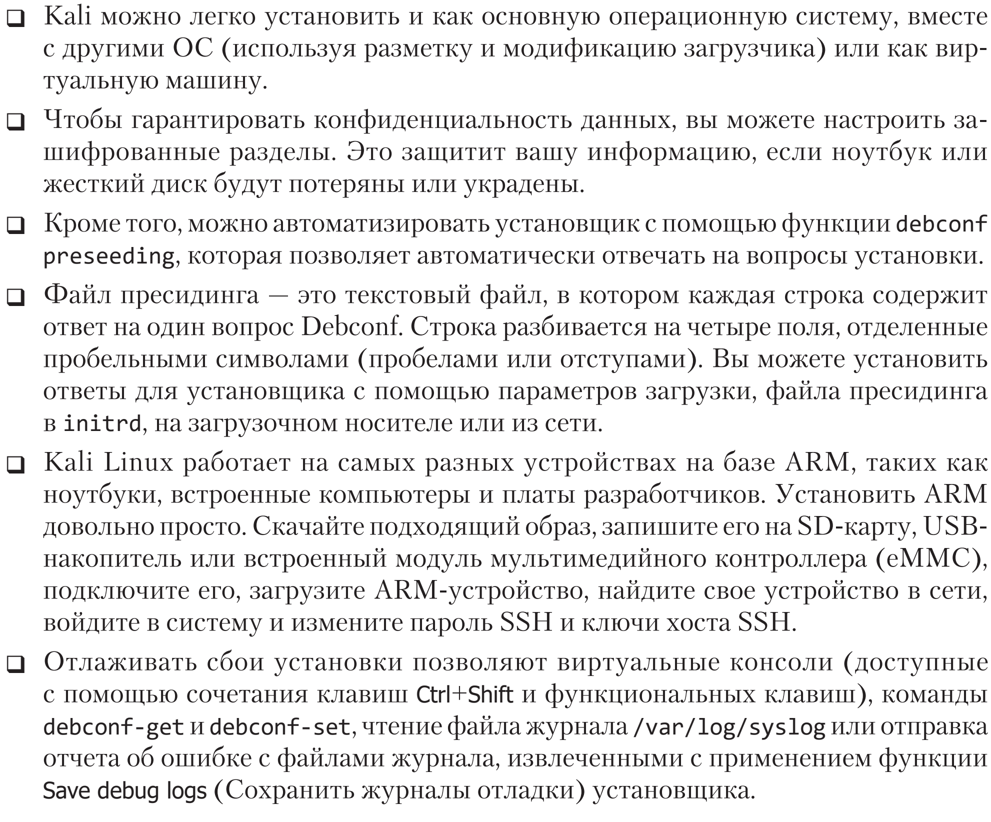


Рис. 4.30. Сохранить журналы отладки (2/2)

4.6. Резюме

В этой главе мы сосредоточились на процессе установки Kali Linux. Обсудили минимальные требования к установке дистрибутива, процесс установки для стандартных и полностью зашифрованных файловых систем. Мы также уделили внимание пресидингу, который позволяет выполнять автоматические установки, инсталляции Kali Linux на различных ARM-устройствах и тому, как действовать в редких случаях сбоя установки.

- Требования к установке для Kali Linux варьируются от базового SSH-сервера без рабочего стола (всего лишь 128 Мбайт ОЗУ (рекомендуется 512 Мбайт) и 2 Гбайта дискового пространства) до более высокоуровневого метапакета kali-linux-full с не менее чем 2048 Мбайт ОЗУ и 20 Гбайт дискового пространства. Кроме того, ваш компьютер должен иметь процессор, поддерживаемый хотя бы одной из архитектур amd64, i386, armel, armhf или arm64.



- Kali можно легко установить и как основную операционную систему, вместе с другими ОС (используя разметку и модификацию загрузчика) или как виртуальную машину.
- Чтобы гарантировать конфиденциальность данных, вы можете настроить зашифрованные разделы. Это защитит вашу информацию, если ноутбук или жесткий диск будут потеряны или украдены.
- Кроме того, можно автоматизировать установщик с помощью функции `debconf preseeding`, которая позволяет автоматически отвечать на вопросы установки.
- Файл пресидинга — это текстовый файл, в котором каждая строка содержит ответ на один вопрос Debconf. Строка разбивается на четыре поля, отделенные пробельными символами (пробелами или отступами). Вы можете установить ответы для установщика с помощью параметров загрузки, файла пресидинга в `initrd`, на загрузочном носителе или из сети.
- Kali Linux работает на самых разных устройствах на базе ARM, таких как ноутбуки, встроенные компьютеры и платы разработчиков. Установить ARM довольно просто. Скачайте подходящий образ, запишите его на SD-карту, USB-накопитель или встроенный модуль мультимедийного контроллера (eMMC), подключите его, загрузите ARM-устройство, найдите свое устройство в сети, войдите в систему и измените пароль SSH и ключи хоста SSH.
- Отлаживать сбои установки позволяют виртуальные консоли (доступные с помощью сочетания клавиш Ctrl+Shift и функциональных клавиш), команды `debconf-get` и `debconf-set`, чтение файла журнала `/var/log/syslog` или отправка отчета об ошибке с файлами журнала, извлеченными с применением функции Save debug logs (Сохранить журналы отладки) установщика.

Теперь, когда мы рассмотрели основы Linux и установку Kali Linux, обсудим конфигурацию, чтобы вы могли приступить к настройке Kali в соответствии с вашими задачами.

ГЛАВА 5

Настройка Kali Linux



Ключевые темы:

- сеть;
- пользователи и группы;
- сервисы;
- Apache;
- PostgreSQL;
- SSH.

В данной главе мы рассмотрим различные способы настройки Kali Linux. Для начала в разделе 5.1 мы продемонстрируем способы настройки параметров сети с помощью графической среды и командной строки. В разделе 5.2 поговорим о пользователях и группах, покажем, как создавать и изменять учетные записи пользователей, устанавливать пароли, отключать учетные записи и управлять группами. Наконец в разделе 5.3 обсудим, что такое сервисы, и объясним, как настроить и поддерживать общие сервисы, а также сосредоточимся на трех очень важных и особых сервисах: SSH, PostgreSQL и Apache.

5.1. Настройка сети

На рабочем столе с помощью инструмента NetworkManager

На стандартном рабочем столе у вас уже установлен инструмент `NetworkManager`, и его можно запускать и настраивать через центр управления GNOME, а также с помощью меню в правом верхнем углу (рис. 5.1).



Рис. 5.1. Экран конфигурации сети

Конфигурация по умолчанию использует DHCP для получения IP-адреса, DNS-сервера и шлюза, но с помощью значка шестеренки в правом нижнем углу можно изменять конфигурацию разными способами (например, установить MAC-адрес, переключиться на статическую настройку, включить или отключить IPv6 и добавить дополнительные маршруты). Вы можете создавать профили для сохранения нескольких конфигураций проводных сетей и легко переключаться между ними. Настройки беспроводных сетей автоматически привязаны к их общедоступному идентификатору (SSID).

`NetworkManager` также обрабатывает соединения с помощью мобильного широкополосного доступа (Wireless Wide Area Network, WWAN) и модемов, использующих двухточечный протокол через Ethernet (PPPoE). И последнее, но не менее важное: `NetworkManager` обеспечивает интеграцию со многими типами виртуальных частных сетей (VPN) через специализированные модули: SSH, OpenVPN, VPNC Cisco, PPTP, Strongswan. Проверьте пакеты `network-manager-*`; большинство из них не установлены по умолчанию. Обратите внимание, что вам нужны пакеты с суффиксом `-gnome`, чтобы иметь возможность настроить их через графический интерфейс пользователя.

В командной строке с помощью пакета Ifupdown

В качестве альтернативы, когда вы предпочитаете не использовать графический рабочий стол (или не имеете к нему доступа), можете настроить сеть с помощью уже установленного пакета `ifupdown`, который включает инструменты `ifup` и `ifdown`. Они считывают определения из файла конфигурации `/etc/network/interfaces` и лежат в основе сценария инициализации `/etc/init.d/networking`, который настраивает сеть во время загрузки.

Каждое сетевое устройство, управляемое `ifupdown`, может быть деконфигурировано в любое время с помощью команды `ifdown` *`сетевое_устройство`*. Затем вы можете изменить файл `/etc/network/interfaces` и запустить сеть (с новой конфигурацией) через команду `ifup` *`сетевое_устройство`*.

Посмотрим, что мы можем добавить в конфигурационный файл `ifupdown`. Существует две основные директивы: автоматическое *сетевое устройство*, которое сообщает `ifupdown` о необходимости автоматически настроить сетевой интерфейс, как только он будет доступен, и *тип inet/inet6 сетевого устройства* `iface` для настройки данного интерфейса. Например, простая конфигурация DHCP выглядит так:

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet dhcp
```

Обратите внимание: в этом файле всегда должна присутствовать специальная конфигурация для устройства `loopback`. Для конфигурации фиксированного IP-адреса

необходимо предоставить более подробную информацию, такую как IP-адрес, сеть и IP-адрес шлюза:

```
auto eth0
iface eth0 inet static
  address 192.168.0.3
  netmask 255.255.255.0
  broadcast 192.168.0.255
  network 192.168.0.0
  gateway 192.168.0.1
```

Для беспроводных интерфейсов вам необходим пакет `wpasupplicant` (по умолчанию включен в Kali), предоставляющий множество параметров `wpa-*`, которые могут использоваться в файле `/etc/network/interfaces`. Ознакомьтесь с файлом `/usr/share/doc/wpasupplicant/README.Debian.gz` для получения примеров и объяснений. Наиболее распространенные параметры — `wpa-ssid` (определяющий имя соединяемой беспроводной сети) и `wpa-psk` (определяющий пароль или ключ, защищающий сеть).

```
iface wlan0 inet dhcp
  wpa-ssid MyNetWork
  wpa-psk plaintextsecret
```

В командной строке с помощью инструмента systemd-networkd

Несмотря на то что `ifupdown` — это древний инструмент, используемый Debian, и хотя он по-прежнему выступает как программа по умолчанию для серверов или других минимальных установок, существует новый инструмент, заслуживающий внимания, — `systemd-networkd`. Интеграция с системой инициализации `systemd` делает его очень привлекательным. Он не относится к дистрибутивам на основе Debian (в отличие от `ifupdown`) и является очень маленьким, эффективным и относительно легким в настройке при условии, что вам понятен синтаксис файлов `unitd`. Он особенно предпочтителен, если вы считаете, что `NetworkManager` излишне наворочен и его сложно настраивать.

Для настройки `systemd-networkd` нужно поместить сетевые файлы в каталог `/etc/systemd/network/` (примеры 5.1, 5.2). Кроме того, вы можете использовать расположение `/lib/systemd/network/` для пакетированных файлов или `/run/systemd/network/` для файлов, сгенерированных во время выполнения. Формат этих файлов документируется в `systemd.network(5)`. Раздел `Match` указывает сетевые интерфейсы, к которым применяется конфигурация. Вы можете задать интерфейс различными способами, в том числе через адрес управления доступом к среде (MAC) или тип устройства. Раздел `Network` определяет конфигурацию сети.

Пример 5.1. Конфигурация на основе DHCP в /etc/systemd/network/80-dhcp.network

```
[Match]
Name=en*

[Network]
DHCP=yes
```

Пример 5.2. Статическая конфигурация в /etc/systemd/network/50-static.network

```
[Match]
Name=enp2s0

[Network]
Address=192.168.0.15/24
Gateway=192.168.0.1
DNS=8.8.8.8
```

Обратите внимание: `system-networkd` отключен по умолчанию, поэтому при желании его использовать вы должны сначала включить этот инструмент. Он также зависит от системного разрешения для правильной интеграции разрешения DNS, что, в свою очередь, требует замены `/etc/resolv.conf` символьной ссылкой на `/run/system/resolve/resolv.conf`, управляемой `systemd-resolved`.

```
# systemctl enable systemd-networkd
# systemctl enable systemd-resolved
# systemctl start systemd-networkd
# systemctl start systemd-resolved
# ln -sf /run/system/resolve/resolv.conf /etc/resolv.conf
```

Хотя `systemd-networkd` и присущи некоторые ограничения, например отсутствие интегрированной поддержки беспроводных сетей, вы можете полагаться на ранее существовавшую внешнюю конфигурацию `wpa_supplicant` для беспроводной поддержки. Однако данный инструмент особенно полезен в контейнерах и виртуальных машинах и был первоначально разработан для сред, в которых конфигурация сети контейнера зависела от конфигурации сети хоста. В этом случае `systemd-networkd` упрощает управление обеими сторонами последовательно, поддерживая всевозможные виртуальные сетевые устройства, которые могут потребоваться в таком типе сценариев (см. `systemd.netdev(5)`).

5.2. Управление пользователями и группами Unix

База данных пользователей и групп Unix состоит из текстовых файлов `/etc/passwd` (список пользователей), `/etc/shadow` (зашифрованные пароли пользователей), `/etc/group` (список групп) и `/etc/gshadow` (зашифрованные пароли групп). Их форматы задокументированы в `passwd(5)`, `shadow(5)`, `group(5)` и `gshadow(5)` соответственно. Хотя эти файлы можно отредактировать вручную с помощью таких инструментов, как `vipw` и `vigr`, выполнять наиболее распространенные операции позволяют средства более высокого уровня.

Создание учетных записей пользователей

Хотя Kali чаще всего запускается при аутентификации в качестве пользователя `root`, вам зачастую по разным причинам понадобится создавать незащищенные учетные записи пользователей, особенно если вы применяете Kali в качестве основной операционной системы. Наиболее типичный способ добавить пользователя — ввести

команду `adduser`, принимающую в качестве аргумента имя нового пользователя, которого вы хотите создать.

Команда `adduser` задает несколько вопросов перед созданием учетной записи, но работать с ней довольно просто. Ее конфигурационный файл `/etc/adduser.conf` содержит множество интересных настроек. Вы можете, например, назначить диапазон идентификаторов пользователей (UID), которые можно применять, указать, входят ли пользователи в общую группу, определить оболочку по умолчанию и т. д.

Создание учетной записи приводит к появлению личного каталога пользователя с содержимым `/etc/skel/template`. В ней пользователю предоставляется набор стандартных каталогов и файлов конфигурации.

В некоторых случаях будет полезно добавить пользователя в группу (отличную от основной группы по умолчанию), чтобы предоставить дополнительные разрешения. Например, пользователь, включенный в группу `sudo`, имеет полные административные привилегии с помощью команды `sudo`. Добавление в группу осуществляется через команду `adduser` *`пользователь группа`*.

Применение команды getent для консультации с пользовательской базой данных

Команда getent (get entries — получить записи) проверяет системные базы данных (в том числе пользователей и групп) с применением соответствующих функций библиотеки, которые, в свою очередь, вызывают модули коммутатора имен (NSS), настроенные в файле /etc/nsswitch.conf. Команда принимает один или два аргумента: имя проверяемой базы и возможный ключ поиска. Таким образом, команда getent passwd kaliuser1 вернет информацию из пользовательской базы данных относительно пользователя kaliuser1.

```
root@kali:~# getent passwd kaliuser1
kaliuser1:x:1001:1001:Kali User,4444,123-867-5309,321-867-
    ➥5309:/home/kaliuser1:/bin/bash
```

Изменение существующей учетной записи или пароля

Следующие команды позволяют изменять информацию, хранящуюся в определенных полях пользовательских баз данных:

- `passwd` — позволяет обычным пользователям изменять свой пароль, что, в свою очередь, обновляет файл `/etc/shadow`;
- `chfn` (Change Full Name — сменить полное имя) — зарезервировано для суперпользователя (`root`), изменяет поле `GECOS` («общая информация»);
- `chsh` (CHange Shell — сменить оболочку) — изменяет оболочку входа пользователя. Однако доступные варианты будут ограничены теми, которые перечислены в файле `/etc/shells`; администратор, с другой стороны, не связан этим ограничением и может установить оболочку к любой выбранной программе;
- `chage` (CHange AGE — сменить срок) — позволяет администратору изменять настройки срока действия пароля, передавая имя пользователя в качестве аргумента, или отображает текущие настройки с помощью параметра `-l` *`пользователь`*.

Кроме того, вы можете принудительно завершить срок действия пароля с помощью команды `passwd -e` *пользователь*, которая заставит пользователя сменить пароль при следующем входе в систему.

Отключение учетной записи

Вам может потребоваться отключить учетную запись (заблокировать пользователя) в качестве дисциплинарной меры, в целях расследования или просто в случае длительного или окончательного отсутствия пользователя. Отключенная учетная запись означает, что пользователь не может войти в систему или получить доступ к компьютеру. Учетная запись остается неповрежденной на машине, и никакие файлы или данные не удаляются; она просто недоступна. Это достигается с помощью команды `passwd -l` *пользователь* (lock — «заблокировать»). Повторное включение учетной записи выполняется аналогичным образом благодаря параметру `-u` (unlock — «разблокировать»).

Управление Unix-группами

Команды `addgroup` и `delgroup` добавляют или удаляют группу соответственно. Команда `groupmod` изменяет информацию группы (ее `gid` или идентификатор). Команда `gpasswd`*группа* изменяет пароль для группы, а команда `gpasswd -r` *группа* удаляет ее.

Работа с несколькими группами

Каждый пользователь может быть членом нескольких групп. Основная группа по умолчанию создается во время начальной настройки пользователя. Изначально каждый файл, созданный пользователем, принадлежит ему, а также его основной группе. Это не всегда желательно; например, когда пользователь должен работать в каталоге, который применяется группой, отличной от его основной группы. В таком случае пользователю необходимо сменить группу с помощью одной из следующих команд: newgrp, которая запускает новую оболочку, или sg, которая просто выполняет команду, задействуя предоставленную альтернативную группу. Эти команды также позволяют пользователю присоединиться к группе, к которой он в настоящее время не принадлежит. Если группа защищена паролем, то необходимо будет указать соответствующий пароль перед выполнением команды.

В качестве альтернативы пользователь может установить бит setgid в каталоге, благодаря которому файлы, созданные в этом каталоге, будут автоматически принадлежать к правильной группе. Для получения дополнительной информации см. выше врезку «Каталог setgid и липкий бит».

Команда id отображает текущее состояние пользователя, его личный идентификатор (переменная uid), текущую основную группу (переменная gid) и список групп, к которым он принадлежит (переменная group).

5.3. Настройка сервисов

В этом разделе мы рассмотрим сервисы (иногда называемые демонами), или программы, которые запускаются в фоновом режиме и выполняют различные системные функции. Мы начнем с обсуждения конфигурационных файлов и продолжим объяснением того, как работают некоторые важные сервисы (такие как SSH, PostgreSQL и Apache) и как их можно настроить.

Настройка конкретной программы

Если вы хотите настроить неизвестный пакет, то должны действовать поэтапно. Во-первых, стоит прочесть, что было задокументировано сопровождающим пакета. Файл `/usr/share/doc/`*`пакет`*`/README.Debian` — хорошее начало. Он часто содержит информацию о пакете, в том числе указатели, которые могут ссылаться на другую документацию. Вы сэкономите много времени и избежите большого количества проблем, сначала прочитав этот файл, поскольку он подробно описывает наиболее популярные ошибки и содержит решения большинства распространенных проблем.

Затем вы должны ознакомиться с официальной документацией программного обеспечения. Обратитесь к разделу 6.1, чтобы узнать, как найти различные источники. Команда `dpkg -L` *`пакет`* предоставляет список файлов, включенных в пакет; с ее помощью вы сможете быстро обнаружить имеющуюся документацию (а также файлы конфигурации, расположенные в `/etc/`). Кроме того, команда `dpkg -s` *`пакет`* отображает метаданные пакета и показывает любые потенциальные рекомендуемые или предлагаемые пакеты; там вы можете найти документацию или, вероятно, утилиту, которая упростит настройку ПО.

Наконец, файлы конфигурации часто самодокументируются многими пояснительными комментариями, в которых описываются различные вероятные значения для каждого параметра конфигурации. В одних случаях вы можете запускать программное обеспечение, раскомментировав одну строку в файле конфигурации. В других случаях примеры файлов конфигурации содержатся в каталоге `/usr/share/doc/`*`пакет`*`/examples/`. Они могут служить основой для вашего собственного файла конфигурации.

Настройка SSH для удаленного входа в систему

SSH позволяет удаленно входить в систему, передавать файлы или выполнять команды. Это стандартный отраслевой инструмент (`ssh`) и сервис (`sshd`) для удаленного подключения к машинам.

Хотя пакет `openssh-server` установлен по умолчанию, сервис `SSH` по умолчанию отключен и, таким образом, не запускается во время загрузки. Вы можете вручную запустить этот сервис с помощью команды `systemctl start ssh` или

настроить его запуск во время загрузки, воспользовавшись командой `systemctl enable ssh`.

Сервис SSH имеет относительно нормальную конфигурацию по умолчанию, но, учитывая его широкие возможности и тонкости настройки, полезно знать, что вы можете сделать с его конфигурационным файлом, `/etc/ssh/sshd_ config`. Все параметры задокументированы в `sshd_config(5)`.

Конфигурация по умолчанию отключает логины с паролями для пользователя `root`, а это значит, что вы должны сначала настроить SSH-ключи с помощью `ssh-keygen`. Вы можете распространить это на всех пользователей, присвоив параметру `PasswordAuthentication` значение `no`, или снять данное ограничение, присвоив параметру `PermitRootLogin` значение `yes` (вместо стандартного запрета-пароля). Сервис `SSH` по умолчанию прослушивает порт 22, но вы можете изменить его с помощью директивы `Port`.

Чтобы применить новые настройки, следует выполнить команду `systemctl reload ssh`.

Создание новых хост-ключей SSH

Каждый SSH-сервер имеет собственные криптографические ключи; они называются хост-ключами SSH и хранятся в виде /etc/ssh/ssh_host_*. Они должны быть приватными, если вам нужна конфиденциальность, и их нельзя использовать на нескольких компьютерах одновременно.

При установке системы путем копирования полного образа диска (вместо использования debian-installer) образ может содержать предварительно сгенерированные хост-ключи SSH, которые вы должны заменить новыми ключами. Вероятно, образ также содержит пароль root по умолчанию, который вы захотите сбросить. Все это можно сделать с помощью следующих команд:

```
# passwd
[...]
# rm /etc/ssh/ssh_host_*
# dpkg-reconfigure openssh-server
# service ssh restart
```

Настройка баз данных PostgreSQL

PostgreSQL — сервер баз данных. Он редко бывает полезен сам по себе, но используется многими другими сервисами для хранения информации. Эти сервисы обычно получают доступ к серверу базы по сети и, как правило, требуют, чтобы к учетным данным для аутентификации можно было подключиться. Таким образом, для настройки этих сервисов требуется создание баз данных PostgreSQL и учетных записей пользователей с соответствующими правами в базе. Чтобы это сделать, нам необходим запущенный сервис, так что начнем с команды `systemctl start postgresql`.

Поддержка нескольких версий PostgreSQL

Пакет PostgreSQL позволяет совместно устанавливать несколько версий сервера базы данных. Можно также обрабатывать несколько кластеров (кластер представляет собой набор баз, обслуживаемых одним и тем же почтовым мастером). В этих целях файлы конфигурации хранятся по адресу /etc/postgresql/версия/имя-кластера/.

Чтобы кластеры работали бок о бок, каждому новому кластеру присваивается следующий доступный номер порта (обычно 5433 для второго кластера). Файл postgresql.service представляет собой пустую оболочку, что упрощает взаимодействие со всеми кластерами, поскольку каждый из них имеет собственный блок (postgresql@версия-кластера.service).

Тип подключения и аутентификация клиента

По умолчанию PostgreSQL прослушивает входящие соединения двумя способами: на TCP-порте `5432` интерфейса локального хоста и файловом сокете `/var/run/postgresql/.s.PGSQL.5432`. Это можно сконфигурировать в файле `postgresql.conf` с различными директивами: `listen_addresses` для адресов для прослушивания, `port` для TCP-порта и `unix_socket_directories`, чтобы определить каталог, в котором созданы сокеты на основе файлов.

Клиенты аутентифицируются по-разному, в зависимости от их подключения. Файл конфигурации `pg_hba.conf` определяет, кому в каком сокете разрешено подключаться и как аутентифицируются клиенты. По умолчанию соединения в файловом сокете применяют учетную запись пользователя Unix в качестве имени пользователя PosgreSQL и предполагают, что дальнейшая проверка подлинности не требуется. TCP-подключение PostgreSQL требует, чтобы пользователь аутентифицировался с именем пользователя и паролем (хотя данное имя пользователя/пароль не Unix, а скорее созданные самим PostgreSQL).

Пользователь `postgres` является специальным и имеет полные административные привилегии по всем базам данных. Мы будем применять его для создания новых пользователей и баз.

Создание пользователей и баз данных

Команда `createuser` добавляет нового пользователя, а `dropuser` удаляет его. Аналогично команда `createdb` добавляет новую базу данных, а `dropdb` удаляет ее. Каждая из этих команд имеет собственные инструкции, но мы рассмотрим здесь некоторые параметры. Каждая команда действует на кластер по умолчанию (работает на порте `5432`), но вы можете задать параметр `--port=`*порт* для изменения пользователей и баз данных альтернативного кластера.

Этим командам необходимо подключаться к серверу PostgreSQL для выполнения своей работы и быть аутентифицированными как пользователь с достаточными полномочиями для совершения указанной операции. Самый простой способ

добиться цели — использовать учетную запись postgres Unix и подключиться к файловому сокету:

```
# su – postgres
$ createuser -P king_phisher
Enter password for new role:
Enter it again:
$ createdb -T template0 -E UTF-8 -O king_phisher king_phisher
$ exit
```

В приведенном выше примере параметр -P дает указание createuser запросить пароль после создания нового пользователя king_phisher. В команде createdb параметр -O определяет пользователя, владеющего новой базой данных (который, таким образом, имеет полные права на создание таблиц, предоставление разрешений и т. д.). Мы также хотим применить строки в Юникоде, поэтому добавляем параметр -E UTF-8 для установки кодировки, что, в свою очередь, требует от нас использовать параметр -T для выбора другого шаблона базы.

Теперь мы можем проверить вероятность подключения к базе данных через сокет, прослушивающий localhost (-h localhost) в качестве пользователя king_phisher (-U king_phisher):

```
# psql -h localhost -U king_phisher king_phisher
Password for user king_phisher:
psql (9.5.2)
SSL connection (protocol: TLSv1.2, cipher: ECDHE-RSA-AES256-GCM-SHA384, bits:
    ➥256, compression: off)
Type "help" for help.

king_phisher=>
```

Как видите, соединение было успешным.

Управление кластерами PostgreSQL

Для начала стоит отметить, что понятие «кластер PostgreSQL» является дополнением, специфическим для Debian, и вы не найдете ссылки на этот термин в официальной документации PostgreSQL. С точки зрения инструментов PostgreSQL, такой кластер представляет собой всего лишь пример сервера базы данных, работающего на определенном порте.

Тем не менее пакет Debian postgresql-common предоставляет несколько инструментов для управления такими кластерами: pg_createcluster, pg_dropcluster, pg_ctlcluster, pg_upgradecluster, pg_renamecluster и pg_lsclusters. Мы не будем описывать все эти инструменты здесь, но вы можете обратиться к соответствующим страницам руководства для получения дополнительной информации.

Вы должны знать, что после установки в вашей системе новая основная версия PostgreSQL создаст новый кластер, который станет работать на следующем порте (обычно 5433), и вы будете продолжать использовать старую версию до тех пор, пока не перенесете свои базы данных из старого кластера в новый.

Вы можете получить список всех кластеров и их статусов с помощью команды `pg_lsclusters`. Более того, есть способ автоматизировать перенос своего кластера на последнюю версию PostgreSQL, используя команду `pg_upgradecluster` *`старая-версия имя-кластера`*. Для этого вам может потребоваться сначала удалить кластер (пустой), созданный для новой версии (с помощью команды `pg_dropcluster` *`новая-версия имя-кластера`*). Старый кластер не удаляется в процессе, но он также не будет запущен автоматически. Вы можете удалить его, как только убедитесь, что обновленный кластер работает исправно.

Настройка сервера Apache

Типичная установка Kali Linux включает веб-сервер Apache, предоставляемый пакетом `apache2`. Будучи сетевым сервисом, он по умолчанию отключен. Вы можете запустить его вручную с помощью команды `systemctl start apache2`.

Поскольку все больше программ выпускается в виде веб-приложений, то важно знать кое-что о применении Apache для размещения этих приложений, будь то локальное использование или общий доступ в сети.

Apache — модульный сервер, и многие функции реализуются внешними модулями, которые загружаются основной программой во время его инициализации. Конфигурация по умолчанию содержит только самые распространенные модули, но включение новых модулей легко осуществляется с помощью команды `a2enmod` *`модуль`*. Примените команду `a2dismod` *`модуль`* для отключения модуля. Эти программы фактически создают (или удаляют) символьные ссылки в `/etc/apache2/mods-enabled/`, указывая на фактические файлы (хранящиеся по адресу `/etc/apache2/mods-available/`).

Существует много доступных модулей, но два из них заслуживают отдельного рассмотрения: PHP и SSL. Веб-приложения, написанные с помощью PHP, выполняются веб-сервером Apache с применением выделенного модуля, предоставляемого пакетом `libapache-mod-php`, и его установка активирует модуль автоматически.

Apache 2.4 включает модуль SSL, необходимый для работы безопасного протокола HTTP (HTTPS) из блока. Сначала его нужно активировать с помощью команды `a2enmod ssl`, затем в файлы конфигурации следует добавить определенные директивы. Пример конфигурации представлен файлом `/etc/apache2/sites-available/default-ssl.conf`. Дополнительную информацию см. на странице http://httpd.apache.org/docs/2.4/mod/mod_ssl.html.

Полный список стандартных модулей Apache можно найти в Интернете по адресу http://httpd.apache.org/docs/2.4/mod/index.html.

По умолчанию веб-сервер прослушивает порт `80` (так указано в файле `/etc/apache2/ports.conf`) и загружает страницы из каталога `/var/www/html/` (как указано в файле `/etc/apache2/sites-enabled/000-default.conf`).

Настройка виртуальных хостов

Виртуальный хост — дополнительная идентификация для веб-сервера. Один и тот же процесс Apache может обслуживать несколько сайтов (например, `https://`

www.kali.org/ и https://www.offensive-security.com/), поскольку HTTP-запросы включают как имя запрашиваемого сайта, так и локальную часть URL (эта функция называется «виртуальный хост, привязанный к имени»).

Конфигурация по умолчанию для Apache 2 активирует виртуальные хосты, привязанные к имени. Кроме того, виртуальный хост по умолчанию определяется в файле /etc/apache2/sites-enabled/000-default.conf; этот виртуальный хост будет использоваться, если не найден хост, соответствующий запросу, отправленному клиентом.

Внимание!



Запросы относительно неизвестных виртуальных хостов всегда будут обслуживаться первым определенным виртуальным хостом, поэтому пакет отправляет файл конфигурации 000-default.conf, в первую очередь сортирующий все другие файлы, которые вы могли создать.

Затем каждый дополнительный виртуальный хост описывается файлом, хранящимся в каталоге /etc/apache2/sites-available/. Обычно файл носит имя хоста сайта, за которым следует суффикс .conf (например: www.example.com.conf). После этого вы можете включить новый виртуальный хост с помощью команды a2ensite www.example.com. Ниже приведена минимальная конфигурация виртуального хоста для сайта, файлы которого хранятся в каталоге /srv/www.example.com/www/ (определяется с применением параметра DocumentRoot):

```
<VirtualHost *:80>
ServerName www.example.com
ServerAlias example.com
DocumentRoot /srv/www.example.com/www
</VirtualHost>
```

Кроме того, можно добавить директивы CustomLog и ErrorLog для настройки в Apache вывода журналов в файлы, предназначенные для виртуального хоста.

Общие директивы

В этом пункте кратко рассматриваются некоторые из обычно используемых конфигурационных настроек Apache.

Основной файл конфигурации, как правило, включает несколько блоков Directory; они позволяют указывать разные типы поведения для сервера в зависимости от местоположения файла, который будет обслуживаться. Такой блок обычно содержит директивы Options и AllowOverride:

```
<Directory /var/www>
Options Includes FollowSymLinks
AllowOverride All
DirectoryIndex index.php index.html index.htm
</Directory>
```

Директива `DirectoryIndex` содержит список файлов, которые нужно проверить, когда клиентский запрос соответствует каталогу. Используется и отправляется в качестве ответа первый существующий файл в списке.

В директиве `Options` следует список параметров, которые можно включить. Значение `None` отключает все параметры; соответственно, `All` включает их все, кроме параметра `MultiView`. Доступны такие параметры:

- `ExecCGI` — указывает, что сценарии CGI могут быть выполнены;
- `FollowSymLinks` — сообщает серверу, что символьным ссылкам можно следовать и ответ должен включать содержимое цели таких ссылок;
- `SymLinksIfOwnerMatch` — также указывает серверу следовать символьным ссылкам, но только если ссылка и ее цель имеют одного и того же владельца;
- `Includes` — содержит *включения на стороне сервера* (Server Side Includes, SSI). Это директивы, встроенные в HTML-страницы и выполняемые динамически для каждого запроса;
- `Indexes` — дает задачу серверу вывести содержимое каталога, если HTTP-запрос, отправленный клиентом, указывает на каталог без индексного файла (то есть когда в этом каталоге не существует файлов, упомянутых директивой `DirectoryIndex`);
- `MultiViews` — включает согласование контента; сервер может использовать это для возврата веб-страницы, соответствующей предпочитаемому языку, указанному в браузере.

Требование аутентификации. В некоторых случаях доступ к части сайта должен быть ограничен и предоставляться только авторизированным пользователям, указавшим имя пользователя и пароль.

Файл `.htaccess` содержит директивы конфигурации Apache, применяемые каждый раз, когда запрос затрагивает элемент из каталога, в котором хранится данный файл. Эти директивы являются рекурсивными, расширяя область видимости для всех подкаталогов.

Большинство директив, которые могут выполняться в блоке `Directory`, также допустимы в файле `.htaccess`. В директиве `AllowOverride` перечислены все параметры, которые можно включить или отключить, задействуя этот файл. Стандартное применение данного параметра — ограничение `ExecCGI`, с помощью которого администратор указывает, какие пользователи могут запускать программы под идентификатором веб-сервера (пользователь `www-data`). В примере 5.3 показан файл `.htaccess`, требующий аутентификации.

Пример 5.3. Файл .htaccess, требующий аутентификации

```
Require valid-user
AuthName "Private directory"
AuthType Basic
AuthUserFile /etc/apache2/authfiles/htpasswd-private
```

> **Базовая аутентификация** Система аутентификации, используемая в приведенном выше примере (Basic), имеет минимальную безопасность, поскольку пароль отправляется в открытом тексте (кодируется только в base64, что является простой кодировкой, а не методом шифрования). Кроме того, следует отметить: документы, защищенные этим механизмом, также проходят через сеть в открытой форме. Если безопасность важна, то весь HTTP-сеанс должен быть зашифрован с помощью протокола Transport Layer Sequence (TLS).

Файл `/etc/apache2/authfiles/htpasswd-private` содержит список пользователей и паролей; он обычно обрабатывается с помощью команды `htpasswd`. Например, для добавления пользователя или изменения пароля служит следующая команда:

```
# htpasswd /etc/apache2/authfiles/htpasswd-private пользователь
New password:
Re-type new password:
Adding password for user пользователь
```

Ограничение доступа. Директива `Require` управляет ограничениями доступа к каталогу (и его подкаталогам, рекурсивно).

Она подходит для ограничения доступа по многим критериям; мы покажем ограничение доступа на основе IP-адреса клиента, но этот процесс можно сделать гораздо более эффективным, особенно если несколько правил `Require` согласованы в блоке `RequireAll`.

Например, вы можете ограничить доступ к локальной сети с помощью следующей директивы:

```
Require ip 192.168.0.0/16
```

5.4. Управление сервисами

В Kali используется система инициализации `systemd`, которая не только отвечает за последовательность загрузки, но и выступает в качестве полнофункционального сервис-менеджера, запускающего и контролирующего сервисы.

Систему `systemd` можно вызывать и контролировать с помощью команды `systemctl`. Без каких-либо аргументов она запускает команду `systemctl list-units`, которая выводит список активных *юнитов*. Если вы запустите команду `systemctl status`, то на экран будет выведен иерархический обзор работающих сервисов. Сравнив оба вывода, вы сразу заметите, что существует несколько видов юнитов и *сервисы* являются лишь одним из них.

Каждый сервис представлен сервисным модулем, который описывается сервисным файлом, обычно переданным в `/lib/systemd/system/` (или `/run/systemd/system/`, или `/etc/systemd/system/`; они перечислены в порядке увеличения важности, последний из них самый важный). Каждый из них может быть изменен

другими файлами `service-name.service.d/*.conf` в том же наборе каталогов. Эти юнит-файлы представляют собой текстовые файлы, формат которых вдохновлен известными файлами `*.ini` в операционной системе Microsoft Windows с парами `ключ=значение`, сгруппированными между заголовками `[`*`раздел`*`]`. Ниже приведен пример файла сервиса для `/lib/systemd/system/ssh.service`:

```
[Unit]
Description=OpenBSD Secure Shell server
After=network.target auditd.service
ConditionPathExists=!/etc/ssh/sshd_not_to_be_run

[Service]
EnvironmentFile=-/etc/default/ssh
ExecStart=/usr/sbin/sshd -D $SSHD_OPTS
ExecReload=/bin/kill -HUP $MAINPID
KillMode=process
Restart=on-failure
RestartPreventExitStatus=255
Type=notify

[Install]
WantedBy=multi-user.target
Alias=sshd.service
```

Целевые юниты — еще одна часть структуры `systemd`. Они представляют желаемое состояние, которое вы хотите достичь, активировав юниты (что означает запущенный сервис в случае юнитов сервисов). Они используются главным образом как способ группировки зависимостей от других юнитов. В момент запуска система активирует юниты, необходимые для достижения цели `default.target` (которая является символьной ссылкой на `graphic.target` и, в свою очередь, зависит от `multi-user.target`). Таким образом, все зависимости этих целей активируются во время загрузки.

Такие зависимости выражаются директивой `Wants` в целевом юните. Но вам не нужно редактировать его для добавления новых зависимостей, вы также можете создать символьную ссылку, указывающую на зависимый юнит в каталоге `/etc/systemd/system/target-name.target.wants/`. Именно это делает команда `systemctl enable` *`foo.сервис`*. При включении сервиса вы даете `systemd` указание добавить зависимость от целей, определенных в записи `WantedBy` раздела `[Install]` файла юнита сервиса. И наоборот, `systemctl disable` *`foo.сервис`* сбрасывает символьную ссылку и, следовательно, зависимость.

Команды `enable` и `disable` ничего не меняют в отношении текущего состояния сервисов. Они влияют только на то, что произойдет при следующей загрузке. Если вы хотите запустить сервис немедленно, то выполните команду `systemctl start` *`foo.сервис`*. И наоборот, команда `systemctl stop` *`foo.сервис`* остановит его. Вы также можете проверить текущий статус сервиса, применив команду `systemctl status` *`foo.сервис`*, которая включает последние строки соответствующего журнала. После изменения конфигурации сервиса вы можете перезагрузить или пере-

запустить его: эти операции выполняются с помощью команд `systemctl reload` *`foo.сервис`* и `systemctl restart` *`foo.сервис`* соответственно.

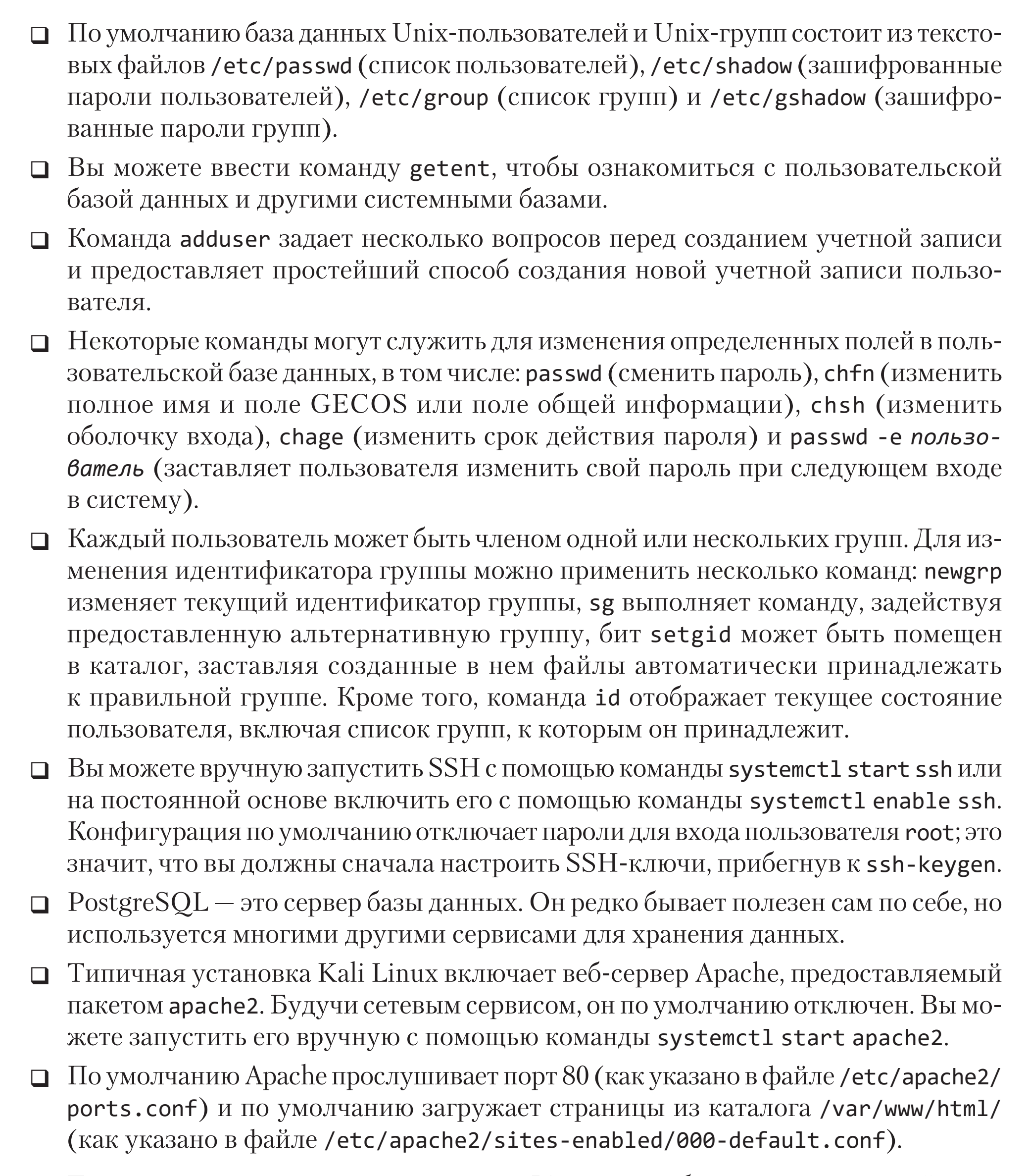
```
# systemctl status postgresql
● postgresql.service – PostgreSQL RDBMS
  Loaded: loaded (/lib/systemd/system/postgresql.service; disabled; vendor
  ➥preset: disabled)
  Active: inactive (dead)
# ls -al /etc/systemd/system/multi-user.target.wants/postgresql.service
ls: cannot access '/etc/systemd/system/multi-user.target.wants/postgresql.
  ➥service': No such file or directory
# systemctl enable postgresql
[...]
# ls -al /etc/systemd/system/multi-user.target.wants/postgresql.service
lrwxrwxrwx 1 root root 38 Apr 21 16:21 /etc/systemd/system/multi-user.target.
  ➥wants/postgresql.service -> /lib/systemd/system/postgresql.service
# systemctl status postgresql
● postgresql.service – PostgreSQL RDBMS
  Loaded: loaded (/lib/systemd/system/postgresql.service; enabled; vendor
  ➥preset: disabled)
  Active: inactive (dead)
# systemctl start postgresql
# systemctl status postgresql
● postgresql.service – PostgreSQL RDBMS
  Loaded: loaded (/lib/systemd/system/postgresql.service; enabled; vendor
  ➥preset: disabled)
  Active: active (exited) since Thu 2016-04-21 16:22:29 EDT; 2s ago
 Process: 6355 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
Main PID: 6355 (code=exited, status=0/SUCCESS)

Apr 21 16:22:29 kali-rolling systemd[1]: Starting PostgreSQL RDBMS...
Apr 21 16:22:29 kali-rolling systemd[1]: Started PostgreSQL RDBMS.
```

5.5. Резюме

В этой главе вы узнали, как настроить Kali Linux. Мы настроили параметры сети, поговорили о пользователях и группах и обсудили, как создавать и изменять учетные записи пользователей, устанавливать пароли, отключать учетные записи и управлять группами. Наконец, мы представили сервисы и объяснили, как настраивать и поддерживать общие сервисы, в частности SSH, PostgreSQL и Apache.

- При обычной установке на рабочем столе у вас уже установлен инструмент `NetworkManager`, и его можно запускать и настраивать через центр управления GNOME и с помощью меню в правом верхнем углу.
- Вы можете настроить сеть из командной строки с помощью инструментов `ifup` и `ifdown`, которые считывают инструкции из файла конфигурации `/etc/network/interfaces`. Еще более новый инструмент — `systemd-networkd` — работает с системой инициализации `systemd`.



- По умолчанию база данных Unix-пользователей и Unix-групп состоит из текстовых файлов `/etc/passwd` (список пользователей), `/etc/shadow` (зашифрованные пароли пользователей), `/etc/group` (список групп) и `/etc/gshadow` (зашифрованные пароли групп).
- Вы можете ввести команду `getent`, чтобы ознакомиться с пользовательской базой данных и другими системными базами.
- Команда `adduser` задает несколько вопросов перед созданием учетной записи и предоставляет простейший способ создания новой учетной записи пользователя.
- Некоторые команды могут служить для изменения определенных полей в пользовательской базе данных, в том числе: `passwd` (сменить пароль), `chfn` (изменить полное имя и поле GECOS или поле общей информации), `chsh` (изменить оболочку входа), `chage` (изменить срок действия пароля) и `passwd -e` *`пользователь`* (заставляет пользователя изменить свой пароль при следующем входе в систему).
- Каждый пользователь может быть членом одной или нескольких групп. Для изменения идентификатора группы можно применить несколько команд: `newgrp` изменяет текущий идентификатор группы, `sg` выполняет команду, задействуя предоставленную альтернативную группу, бит `setgid` может быть помещен в каталог, заставляя созданные в нем файлы автоматически принадлежать к правильной группе. Кроме того, команда `id` отображает текущее состояние пользователя, включая список групп, к которым он принадлежит.
- Вы можете вручную запустить SSH с помощью команды `systemctl start ssh` или на постоянной основе включить его с помощью команды `systemctl enable ssh`. Конфигурация по умолчанию отключает пароли для входа пользователя `root`; это значит, что вы должны сначала настроить SSH-ключи, прибегнув к `ssh-keygen`.
- PostgreSQL — это сервер базы данных. Он редко бывает полезен сам по себе, но используется многими другими сервисами для хранения данных.
- Типичная установка Kali Linux включает веб-сервер Apache, предоставляемый пакетом `apache2`. Будучи сетевым сервисом, он по умолчанию отключен. Вы можете запустить его вручную с помощью команды `systemctl start apache2`.
- По умолчанию Apache прослушивает порт 80 (как указано в файле `/etc/apache2/ports.conf`) и по умолчанию загружает страницы из каталога `/var/www/html/` (как указано в файле `/etc/apache2/sites-enabled/000-default.conf`).

Теперь, когда мы рассмотрели основы Linux и разобрались с установкой и настройкой Kali Linux, обсудим, как устранять неисправности Kali, и изучим некоторые инструменты и приемы, чтобы выполнить резервное копирование и запуск при возникновении проблем.

ГЛАВА



Самостоятельное решение проблем и получение помощи



Ключевые темы:

- документация;
- форумы;
- IRC-канал;
- отчеты об ошибках.

Независимо от вашего опыта, без сомнений, рано или поздно вы столкнетесь с трудностями. Решение проблемы зачастую лежит в ее понимании и в использовании различных ресурсов для поиска «лекарства» и проработки.

В этой главе мы обсудим различные доступные источники информации и рассмотрим лучшие стратегии для поиска необходимой помощи или решения проблем, с которыми вы можете столкнуться. Мы также познакомим вас с некоторыми доступными ресурсами сообщества Kali Linux, включая веб-форумы и сервис интернет-ретрансляций (IRC). Наконец, представим отчет об ошибках и покажем, как использовать системы регистрации ошибок для устранения проблем, изложим стратегии, которые помогут вам создать собственный отчет об ошибке и таким образом быстро и эффективно справляться с недокументированными проблемами.

6.1. Источники документации

Прежде чем вы сможете понять, что же на самом деле происходит при возникновении проблемы, вам необходимо знать теоретическую роль каждой программы, связанной с проблемой. Один из лучших способов сделать это — ознакомиться с документацией программы. Начнем с обсуждения того, где именно вы можете найти документацию, поскольку она часто рассредоточена.

Как избежать ответов типа RTFM

Данная аббревиатура означает «читайте долбаную инструкцию (read the f***ing manual)», но ее можно также расшифровать как более дружественную версию — «обратитесь к прилагаемому руководству (read the fine manual)». Эта фраза иногда используется в качестве (краткого) ответа на вопросы новичков. Она довольно резкая и выдает определенное раздражение, вызываемое вопросом от человека, который даже не удосужился прочесть документацию. Кое-кто считает: этот классический ответ все же лучше, чем ничего, поскольку он хотя бы намекает, что ответ на вопрос кроется в документации.

Когда вы публикуете вопросы, не стоит обижаться на случайный ответ RTFM, но очень важно показать, что вы потратили время на поиски ответа, прежде чем опубликовали вопрос; укажите источники, с которыми вы консультировались, и пошагово опишите свои попытки найти информацию. Все описанные действия помогут продемонстрировать, что вы не ленивы и действительно тяготеете к знаниям. Следование рекомендациям Эрика Раймонда (http://catb.org/~esr/faqs/smart-questions.html) — это хороший способ избежать наиболее распространенных ошибок и получить полезные ответы.

Руководства

Руководства, будучи весьма краткими по объему, содержат немало полезной информации. Чтобы просмотреть определенное руководство, просто введите команду `man` *руководство*. Имя руководства обычно совпадает с именем команды. Так, чтобы узнать подробнее о команде `cp`, вы должны ввести в оболочке командной строки команду `man cp`.

Руководства содержат документацию не только о программах, доступных из командной строки, но и о файлах конфигурации, системных вызовах, функциях библиотеки C и т. д. Иногда имена могут дублироваться. Например, команда `read` для командной строки имеет то же имя, что и системный вызов `read`. Вот почему руководства организованы в следующие пронумерованные разделы.

1. Команды, которые могут быть выполнены из командной строки.
2. Системные вызовы (функции, предоставляемые ядром).
3. Функции библиотек (предоставляемые системными библиотеками).
4. Устройства (в Unix-подобных системах это специальные файлы, обычно помещенные в каталог `/dev/`).
5. Файлы конфигурации (форматы и условные обозначения).
6. Игры.
7. Наборы макросов и стандартов.
8. Команды администрирования системы.
9. Подпрограммы ядра.

Вы можете указать раздел для руководства, которое ищете: для просмотра документации, касающейся системного вызова `read`, вы должны ввести `man 2 read`. Если ни один раздел не указан явно, то будет показан первый раздел, содержащий руководство с запрошенным именем. Таким образом, `man shadow` возвращает `shadow (5)`, поскольку в разделах 1–4 нет руководств для `shadow`.

Конечно, руководство будет не слишком полезным, если вы не знаете имена команд. Введите команду `apropos`, которая ищет руководства (или, точнее, их краткие описания) для любых ключевых слов, предоставленных вами. Эта команда возвращает список руководств, в описании которых упоминаются запрошенные ключевые слова, а также однострочное описание каждого руководства. Если вы правильно выбрали ключевые слова, то сможете найти имя нужной вам команды.

Пример 6.1. Поиск команды cp с помощью команды apropos

```
$  apropos "copy file"
cp (1)               - copy files and directories
cpio  (1)            - copy files  to  and  from  archives
gvfs-copy (1)        - Copy files
gvfs-move  (1)       - Copy files
hcopy  (1)           - copy files  from  or  to  an  HFS  volume
install (1)          - copy files and  set attributes
ntfscp  (8)          - copy file  to  an  NTFS  volume.
```

Просмотр документации с переходом по ссылкам Во многих руководствах, обычно ближе к концу документа, есть раздел See Also, который содержит ссылки на другие руководства, относящиеся к аналогичным командам, или к внешней документации. Вы можете использовать этот раздел, чтобы найти необходимую документацию, даже если первый открытый документ не подошел.

Помимо команды `man` вы можете использовать для поиска руководств команды `konqueror` (в KDE) и `yelp` (в GNOME).

Документы формата info

Проект GNU подготовил инструкции для большинства своих программ в формате `info`; поэтому многие руководства ссылаются на соответствующую документацию данного формата. Он дает некоторые преимущества, но программа по умолчанию для просмотра этих документов (которая также называется `info`) довольно сложная. Мы рекомендуем использовать вместо нее программу `pinfo` (из пакета `pinfo`). Чтобы установить ее, выполните команду `apt update`, а затем `apt install pinfo` (см. пункт «Установка пакетов с помощью APT» подраздела «Установка пакетов» раздела 8.2).

Документация формата `info` имеет иерархическую структуру, и если вы вызовете функцию `pinfo` без дополнительных параметров, то она отобразит список узлов, доступных на первом уровне. Обычно узлы носят имена соответствующих команд.

Вы можете задействовать клавиши со стрелками для перемещения между узлами, а также графический браузер (который намного удобнее для пользователя), например `konqueror` или `yelp`.

Что касается перевода на другие языки, то система `info` всегда выпускается на английском языке и не включает никаких переводов, в отличие от системы руководств `man`. Однако если вы попросите программу `pinfo` отобразить несуществующий документ типа `info`, она вернет вам руководство с тем же именем (при условии, что оно существует), которое можно перевести на нужный вам язык.

Документация для пакетов

Каждый пакет содержит в себе сопутствующую документацию, и даже к наименее документированным программам обычно прилагается файл `README`, хранящий интересную и/или важную информацию. Эта документация размещена в каталоге `/usr/share/doc/`*`пакет`*`/` (где *`пакет`* представляет собой имя пакета). Если документация слишком велика, то может не включаться в основной пакет программы, а быть выгружена в отдельный пакет, который обычно называется *`пакет`*`-doc`. Основной пакет, как правило, ссылается на пакет документации, помогая легко его найти.

В каталоге `/usr/share/doc/`*`пакет`*`/` также содержится ряд файлов, предоставленных Debian, которые дополняют документацию, указывая на особенности или

усовершенствования пакета по сравнению с базовой установкой программного обеспечения. Файл `README.Debian` тоже показывает все коррективы, внесенные в соответствии с политикой Debian. Файл `changelog.Debian.gz` позволяет пользователю отслеживать изменения, внесенные в пакет с течением времени; это поможет понять разницу между двумя установленными версиями, чье поведение различается. Наконец, иногда встречается файл `NEWS.Debian.gz`, в котором задокументированы основные изменения в программе, предназначенные непосредственно для администратора.

Сайты

Нередко вы можете найти сайты, которые служат для распространения бесплатного ПО и общения его разработчиков и пользователей. Эти сайты наполняются соответствующей информацией в различных формах, таких как официальная документация, часто задаваемые вопросы (frequently asked questions, FAQ) и архивы рассылок. В большинстве случаев проблемы, с которыми вы столкнулись, рассматриваются в FAQ или в этих архивах. При поиске информации в Интернете очень важно изучить синтаксис поиска. Небольшой совет: попробуйте ограничить поиск определенным доменом, относящимся к проблемной программе. Если поиск возвращает слишком много страниц или результаты не соответствуют тому, что вы ищете, то можете добавить ключевое слово `kali` или `debian` для ограничения результатов и уточнения искомых сведений.

От ошибки к решению

Если ПО возвращает очень специфическое сообщение об ошибке, то введите его в поисковой системе (в двойных кавычках, ", чтобы искать полную фразу, а не отдельные ключевые слова). В большинстве случаев первые ссылки результата будут содержать нужный ответ.

В других случаях вы можете встретить общие ошибки, такие как Permission denied (Доступ отклонен). В этом случае лучше проверить разрешения для задействованных элементов (файлов, идентификаторов пользователей, групп и т. д.). Короче говоря, не привыкайте постоянно обращаться к поисковой системе, чтобы найти решение проблемы. Вы рискуете легко забыть о том, что здравый смысл никто не отменял.

Если вы не знаете адрес сайта программного обеспечения, то можете воспользоваться одним из способов его определения. Попробуйте для начала найти поле `Homepage` в метаинформации пакета (`apt show` *пакет*). Либо же описание пакета может включать ссылку на официальный сайт программы. Если URL не указан, то он может содержаться в сопровождающей документации пакета в файле `/usr/share/doc/`*пакет*`/copyright`. Наконец, для поиска сайта ПО можно задействовать поисковую систему (например, Google, DuckDuckGo, Yahoo и т. д.).

Документация на сайте docs.kali.org

Проект Kali содержит сборник полезной документации по адресу https://docs.kali.org/. Хотя эта книга охватывает значительную часть того, что вы должны знать о Kali Linux, упомянутая документация может оказаться весьма кстати, поскольку содержит пошаговые инструкции (преимущественно практические руководства) по многим темам.

Рассмотрим представленные на сайте темы:

- начало работы — ряд инструкций, включая инструкции по скачиванию, для тех, кто не знаком с Kali;
- Kali Linux Live — документация, описывающая использование дистрибутива в качестве live-системы;
- установка Kali Linux — различные документы, описывающие установку, в том числе способы установки совместно с другими операционными системами;
- Kali Linux на ARM — множество инструкций по запуску дистрибутива на различных устройствах на базе ARM;
- использование Kali Linux — множество руководств, охватывающих многие распространенные запросы;
- настройка Kali Linux — инструкции для умельцев, желающих изменить Kali в соответствии со своими потребностями;
- поддержка сообщества Kali — ссылки на различные сообщества, где вы можете получить поддержку, и пояснения, как отправлять отчеты об ошибках;
- политика Kali Linux — объяснения особенностей дистрибутива, отличающих его от других дистрибутивов Linux;
- The Kali Linux Dojo — видеозаписи с семинаров Black Hat и DEF CON.

6.2. Сообщества Kali Linux

Во всем мире существует множество сообществ Kali Linux, использующих различные инструменты для общения (например, форумы и социальные сети). В этом разделе мы рассмотрим только два официальных сообщества дистрибутива.

Веб-форумы на сайте forums.kali.org

Официальные форумы сообщества для проекта Kali Linux находятся на сайте https://forums.kali.org/. Как и на любом веб-форуме, сначала нужно создать учетную запись, чтобы иметь возможность публиковать сообщения. Система запоминает, какие сообщения вы просматривали ранее; это позволяет легко следить за разговорами.

Перед публикацией сообщений вы должны ознакомиться с правилами форума (https://docs.kali.org/community/kali-linux-community-forums). Мы не будем дублировать их здесь, но стоит отметить, что там запрещено говорить о нелегальной деятельности, например о проникновении в чужие сети. Вы должны уважать других членов сообщества, помогая создать приветливую атмосферу. Реклама запрещена, также следует избегать обсуждений вне темы. Категорий достаточно для охвата всего, что вы хотели бы обсудить касательно Kali Linux.

Канал IRC #kali-linux в сети Freenode

IRC — это чат-система для общения в реальном времени. Обсуждения происходят в чатах, которые называются каналами, и обычно сосредоточены вокруг определенной темы или сообщества. Проект Kali Linux использует канал `#kali-linux` в сети Freenode (http://freenode.net/) (вы можете применить `chat.freenode.net` в качестве IRC-сервера, задействуя порт 6667 для TLS-соединения или 6666 для соединения с открытым текстом).

Чтобы присоединиться к обсуждениям в IRC, вы должны использовать IRC-клиент `hexchat` (в графическом режиме) или `irssi` (в консольном). Существует также веб-клиент, доступный на сайте http://webchat.freenode.net/.

Хотя присоединиться к разговору очень просто, вы должны знать, что каждый IRC-канал имеет собственные правила и существует оператор канала (его ник содержит префикс `@`), который следит за их соблюдением: он может удалить вас из канала (или даже закрыть доступ к чату, если вы продолжите нарушать правила). Канал `#kali-linux` не исключение. Правила канала описаны здесь: docs.kali.org/community/kali-linux-irc-channel.

Подведем итоги касательно правил: будьте дружелюбны, терпимы и разумны. Избегайте дискуссий вне темы. Кроме того, запрещено обсуждать незаконную деятельность, взломанное/пиратское программное обеспечение, распространяемое бесплатно без разрешения автора, политику и религию. Имейте в виду, что ваш IP-адрес будет доступен другим пользователям.

Если вы хотите обратиться за помощью, то следуйте рекомендациям, приведенным во врезке «Как избежать ответов типа RTFM» ранее в этой главе: сначала попробуйте найти ответ самостоятельно и поделитесь результатами. Когда вас попросят предоставить дополнительную информацию, пожалуйста, предоставьте ее точно (если вы должны показать подробный вывод, то не копируйте его прямо в канал, а используйте службу типа Pastebin (https://pastebin.com/) и отправляйте только URL Pastebin).

Не ждите немедленного ответа. Несмотря на то что IRC — коммуникационная платформа, действующая в режиме реального времени, здесь зарегистрированы участники со всего мира, поэтому часовые пояса и графики работы могут различаться. Для ответа на ваш вопрос может потребоваться несколько минут или часов.

Однако когда кто-то добавит в ответ ваш ник, он будет выделен, и большинство IRC-клиентов пришлет вам соответствующее уведомление, так что оставьте клиент подключенным и будьте терпеливы.

6.3. Подача грамотно составленного отчета об ошибке

Если все ваши попытки решить проблему провалились, то, возможно, проблема связана с программной ошибкой. Результатом такой проблемы должен стать отчет об ошибке. Вероятно, вы сможете найти решение своей проблемы в ранее предоставленных отчетах об ошибках, но рассмотрим процедуру подачи отчета об ошибке в Kali, Debian или непосредственно разработчикам, чтобы понять процесс на случай, если понадобится представить собственный отчет.

Цель отчета об ошибке — предоставить достаточно информации, чтобы разработчики или администраторы (предположительно) неисправной программы могли воспроизвести проблему, отладить ее поведение и исправить. Это значит, что ваш отчет об ошибке должен содержать соответствующую информацию и его нужно направить к правильному человеку или команде проекта. Отчет также должен быть хорошо написанным и подробным, обеспечивая, таким образом, более быстрый ответ.

Точная процедура подачи отчета об ошибке может различаться в зависимости от того, куда вы будете отправлять отчет (Kali, Debian, разработчики), но существуют некоторые общие рекомендации, применимые ко всем случаям. Мы рассмотрим их далее.

Общие рекомендации

Обсудим ряд общих рекомендаций и руководств, способных помочь вам представить отчет об ошибке, который будет понятным, исчерпывающим и увеличит шансы, что разработчики своевременно устранят ошибку.

Как связаться

Напишите свой отчет на английском языке. Сообщество свободного программного обеспечения является международным, и если вы не знаете своего собеседника, то должны использовать английский язык. Если вы носитель этого языка, то стройте простые предложения и избегайте конструкций, которые могут быть трудными для людей со слабыми познаниями в английском. Несмотря на то что большинство разработчиков очень умны, не все из них обладают отличными знаниями английского языка. Лучше на это не рассчитывать.

Уважительно относитесь к деятельности разработчиков. Помните, что большинство разработчиков свободного ПО (включая тех, кто занимается Kali Linux) — добровольцы и тратят свое ограниченное свободное время на работу с программным обеспечением, которое вы используете бесплатно. Многие делают это из альтруизма. Так что когда отправляете отчет об ошибке, проявляйте уважение (даже если ошибка кажется очевидной оплошностью разработчика); не стоит считать, что они вам обязаны. Лучше поблагодарите их за их вклад.

Если вы знаете, как модифицировать и перекомпилировать программное обеспечение, то предложите помочь разработчикам в тестировании любых патчей, которые они вам предоставят. Это покажет, что вы тоже готовы инвестировать свое время в проект.

Будьте активны и готовы предоставить дополнительную информацию. В некоторых случаях разработчик может снова обратиться к вам для получения новых сведений или с просьбой попытаться еще раз воссоздать проблему, допустим, используя другие параметры или обновленный пакет. Постарайтесь ответить как можно быстрее. Чем быстрее вы отвечаете, тем выше вероятность того, что проблема будет решена в скором времени, пока ее первоначальный анализ все еще свеж в памяти разработчиков.

Следует стараться реагировать быстро, но все же не стоит излишне спешить: предоставляемые данные должны быть правильными и содержать все, что запрашивают разработчики. Люди будут раздражены, если им придется просить вас о чем-то во второй раз.

Что включить в отчет об ошибке

Инструкции по воспроизведению проблемы. Чтобы воспроизвести проблему, разработчики должны знать, что вы используете, где вы это взяли и как установили.

Вы должны предоставить точные, пошаговые инструкции, описывающие, как воспроизвести проблему. Если для воспроизведения проблемы необходимы какие-либо дополнительные данные, то прикрепите соответствующий файл к отчету об ошибке. Попытайтесь придумать минимальный набор инструкций, требуемых для воспроизведения ошибки.

Опишите ситуацию и расскажите о своих ожиданиях. Объясните, что именно вы пытались сделать и какого поведения программы ожидали.

В некоторых случаях ошибка возникает только потому, что вы использовали программу не по назначению. Объяснив, чего вы пытались достичь, вы позволите разработчикам определить, такой ли это случай.

В других ситуациях поведение, которое вы описываете как ошибку, на самом деле может быть нормальным. Расскажите подробно, чего вы ожидали от программы. Это прояснит ситуацию для разработчиков. Они могут либо улучшить поведение программы, либо доработать документацию, но по крайней мере узнают, что поведение их программы запутывает некоторых пользователей!

Будьте конкретны. Добавьте номера версий программного обеспечения, которое вы используете, и, при необходимости, номера версий их зависимостей. Когда вы ссылаетесь на что-то скачанное, укажите полный URL.

При получении сообщения об ошибке укажите его в точности так, как оно было выведено. Если возможно, добавьте копию вывода или снимок экрана. Приложите копии всех относящихся к делу файлов журналов, предварительно удалив все конфиденциальные данные.

Упомяните о возможных решениях или способах обхода проблемы. Перед подачей отчета об ошибке вы, вероятно, пробовали решить проблему самостоятельно. Опишите свои попытки и полученные результаты. Объясните предельно ясно, что является фактом, а что было только вашей гипотезой.

Если вы находили в Интернете какие-либо разъяснения о подобной проблеме, то можете их тоже указать, в частности в случае обнаружения других похожих отчетов об ошибках в системе отслеживания ошибок Debian или более высокого уровня.

При нахождении способа достичь желаемого результата без запуска ошибки, пожалуйста, задокументируйте и это. Информация поможет другим пользователям, которые столкнулись с той же проблемой.

Длинные отчеты об ошибках — это нормально. Отчет об ошибках двумя строками является недостаточным; для передачи всей необходимой информации обычно требуется несколько абзацев (а иногда и страниц).

Предоставьте по возможности всю имеющуюся информацию. Постарайтесь сосредоточиться на том, что относится к делу, но если вы не уверены, то лучше больше, чем меньше.

Если ваш отчет об ошибках слишком длинный, то найдите время, чтобы структурировать содержимое, и предоставьте краткое изложение проблемы в начале отчета.

Дополнительные советы

Избегайте подачи одинаковых отчетов об ошибках. В мире свободного программного обеспечения все системы отслеживания ошибок общедоступны. Найденные проблемы можно просмотреть, и есть даже функция поиска среди них. Таким образом, перед подачей нового отчета об ошибке попытайтесь определить, не сообщал ли о вашей проблеме кто-нибудь еще.

При обнаружении существующего отчета об ошибках подпишитесь на него и по возможности добавьте дополнительную информацию. Не стоит писать комментарии типа «Я тоже» или «+1»; они не несут никакой пользы. Но вы можете указать, что готовы к дальнейшим тестированиям, если первоначальный заявитель этого не предлагал.

Если вы не нашли отчета о своей проблеме, то составьте и подайте его. В случае нахождения подходящих тегов обязательно укажите их.

Убедитесь, что используете последнюю версию. Разработчики очень раздражаются, когда получают отчеты об уже исправленных ими ошибках или о проблемах,

которые невозможно воспроизвести с версией ПО, которую они задействуют (а это почти всегда последняя версия продукта). Даже когда старые версии поддерживаются разработчиками, поддержка зачастую ограничивается исправлениями системы безопасности и решением серьезных проблем. Вы уверены, что ваша ошибка одна из них?

Вот почему перед подачей отчета об ошибке вы должны убедиться, что используете последнюю версию проблемной системы и приложения и можете воспроизвести проблему.

Если Kali Linux не предлагает последнюю версию приложения (ни в `kali-rolling`, ни в `kali-bleeding-edge`, см. пункт «Репозиторий Kali-Bleeding-Edge» подраздела «Репозитории Kali»), то можете поступить так: попробуйте установить вручную последнюю версию в виртуальной машине или просмотреть предыдущий файл `ChangeLog` (или историю записей изменений в Git), чтобы убедиться в отсутствии изменений, которые могли бы устранить проблему (и затем зарегистрируйте ошибку, даже если не испытали последнюю версию).

Не смешивайте несколько проблем в одном отчете об ошибке. Создайте отдельный отчет об ошибке для каждой проблемы. Таким образом, последующие обсуждения будут строго организованы и каждая ошибка может быть исправлена в соответствии с ее собственным графиком. Если вы этого не сделаете, то либо одну и ту же ошибку придется исправлять несколько раз и ее можно будет закрыть только после устранения всех проблем из отчета, либо разработчики вынуждены будут подать дополнительные отчеты, которые должны были создать вы.

Где регистрировать отчет об ошибке

Чтобы решить, где зарегистрировать отчет об ошибке, вы должны хорошо понимать проблему и суметь определить, в какой части программного обеспечения она находится.

В идеале вы должны отследить проблему до файла в вашей системе, а затем можете использовать команду `dpkg`, чтобы узнать, к какому пакету принадлежит этот файл и откуда данный пакет. Предположим, вы нашли ошибку в графическом приложении. Просмотрев список запущенных процессов (вывод команды `ps auxf`), вы обнаружили, что приложение было запущено файлом `/usr/bin/sparta`:

```
$ dpkg -S /usr/bin/sparta
sparta: /usr/bin/sparta
$ dpkg -s sparta | grep ^Version:
Version: 1.0.1+git20150729-0kali1
```

Вы видите, что файл `/usr/bin/sparta` предоставляется пакетом `sparta` версии `1.0.1+git 20150729-0kali1`. Содержание в строке версии `kali` указывает на то, что пакет поступил от Kali Linux (или был изменен Kali Linux). Любой пакет, который не содержит слово `kali` в строке версии (или в имени пакета), поступает прямо из Debian (обычно из Debian Testing).

Проверьте дважды, прежде чем регистрировать ошибки в Debian

Если вы обнаружили ошибку в пакете, импортированном напрямую из Debian, то в идеале о ней следует сообщить, и разработчики Debian должны ее исправить. Однако сначала убедитесь в том, что проблема воспроизводима в простой системе Debian, поскольку Kali, возможно, вызвала проблему путем изменения других пакетов или зависимостей.

Самый простой способ проверить это — настроить виртуальную машину, на которой запущен дистрибутив Debian Testing. Вы можете найти ISO-образы Debian Testing на сайте Debian (https://www.debian.org/devel/debian-installer/).

Если проблема воспроизводима на виртуальной машине, то можете отправить отчет об ошибке в Debian, выполнив команду reportbug в виртуальной машине и следуя инструкциям.

Большинство отчетов об ошибках касательно поведения приложений нужно направить к их проектам на более высоком уровне, кроме случаев, когда возникает проблема интеграции: в этом случае ошибка заключается в неправильном пакетировании программного обеспечения и его интеграции в Debian или Kali. Например, если приложение предлагает параметры времени компиляции, которые пакет не допускает, или оно не работает из-за отсутствующей библиотеки (возникает недостающая зависимость в метаинформации пакета), то вы можете столкнуться с проблемой интеграции. Если вы не знаете, с какой именно проблемой вы столкнулись, лучше всего зарегистрировать ошибку с обеих сторон и добавить в отчеты перекрестные ссылки.

Идентифицировать проект более высокого уровня и найти, куда подавать отчет об ошибке, обычно легко. Вам просто нужно просмотреть сайт, ссылка на который содержится в метаданных пакета в поле Homepage:

```
$ dpkg -s sparta | grep ^Homepage:
Homepage: https://github.com/SECFORCE/sparta
```

Как подать отчет об ошибке

Подача отчета об ошибке в Kali

Kali использует сетевую систему отслеживания ошибок, расположенную на сайте https://bugs.kali.org/my_view_page.php, где вы можете просматривать отчеты об ошибках анонимно, но если хотите оставить комментарий или подать новый отчет, то придется создать новую учетную запись.

Регистрация учетной записи в системе отслеживания ошибок. Чтобы начать, щелкните на ссылке Signup for new account (Регистрация новой учетной записи) на сайте системы отслеживания ошибок, как показано на рис. 6.1.

Затем укажите имя пользователя, адрес электронной почты и ответ на вопрос капчи (теста-проверки). Затем нажмите кнопку Signup (Зарегистрироваться), чтобы продолжить (рис. 6.2).

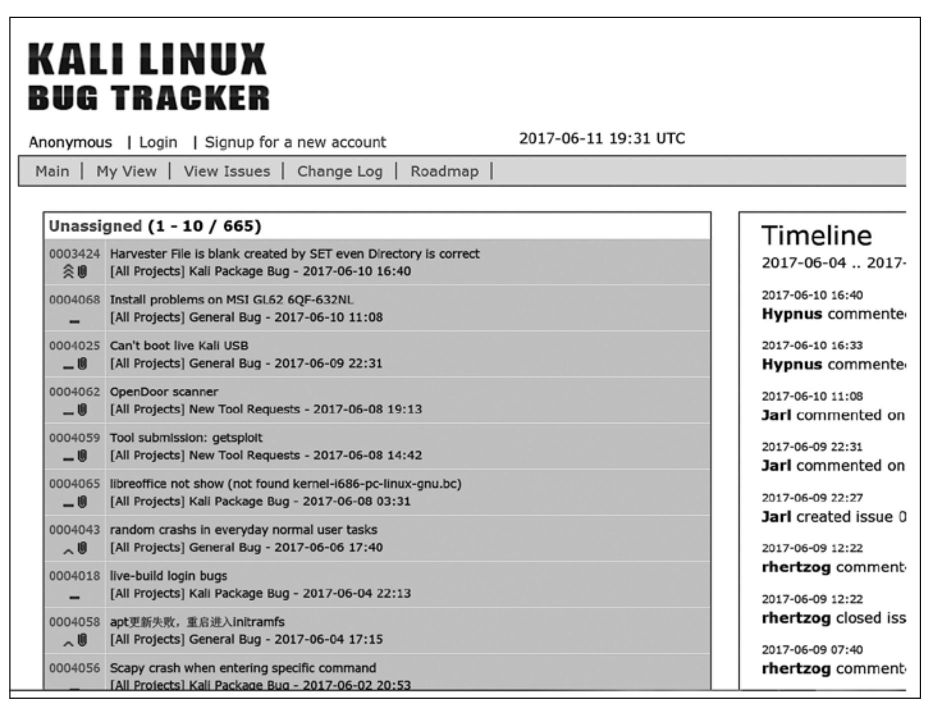


Рис. 6.1. Начальная страница системы отслеживания ошибок Kali

KALI LINUX
BUG TRACKER

Signup	[Login] [Lost your password?]
Username	NewBugSugmitter
E-mail	nbs@email.com
Enter the code as it is shown in the box on the right:	YvRrP [Generate a new code]

On completion of this form and verification of your answers, you will be sent a confirmation message to the e-mail address you specified.

Using the link provided in the e-mail, you will be able to activate your account. If you fail to do so within seven days, it may be purged.

You must specify a valid e-mail address in order to receive the account confirmation e-mail.

Signup

Рис. 6.2. Страница регистрации

При успешной регистрации следующая страница (рис. 6.3) уведомит вас о том, что регистрация учетной записи была обработана, а система отслеживания ошибок отправит письмо с подтверждением на указанный вами адрес. Чтобы активировать учетную запись, нужно будет щелкнуть на ссылке в письме.

KALI LINUX
BUG TRACKER

Account registration
processed.

Congratulations, you have registered successfully ! You are now being sent a confirmation e-mail to verify your e-mail address. Visiting the link sent to you in this e-mail will activate your account.

You have seven days to complete the account confirmation process; if you fail to do so within this period, the newly-registered account may be purged.

[Proceed]

Рис. 6.3. Страница подтверждения регистрации

Как только ваша учетная запись активирована, нажмите кнопку Proceed (Продолжить), чтобы перейти на страницу входа в систему отслеживания ошибок.

Создание отчета. Начать отчет вы сможете, войдя в свою учетную запись и щелкнув на ссылке Report Issue (Сообщить о проблеме), расположенной на исходной странице. Вам будет представлена форма со множеством полей для заполнения (рис. 6.4).

Ниже представлено краткое описание всех полей формы.

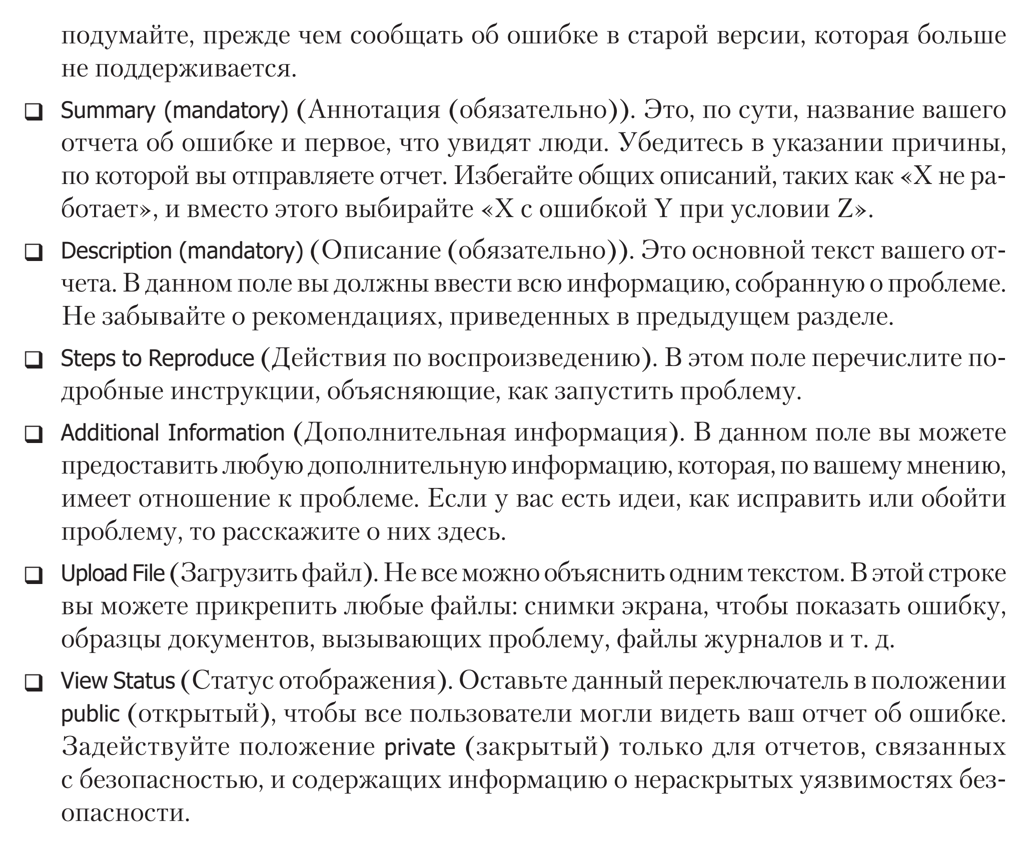
- Category (mandatory) (Категория (обязательно)). В этом раскрывающемся списке выбирается категория сообщаемой ошибки. Отчеты, которые можно отнести к определенному пакету, должны быть представлены в категориях Kali Package Bug (Ошибка пакета Kali) или Kali Package Improvement (Усовершенствование пакета Kali). Для других отчетов нужно использовать категории General Bug (Общая ошибка) или Feature Requests (Требование к функции). Остальные категории предназначены для конкретных случаев: Tool Upgrade (Обновление инструмента) служит для уведомления разработчиков Kali о доступности новой версии

Enter Report Details	
*Category	[All Projects] Kali Package Bug
Reproducibility	have not tried
Severity	minor
Priority	normal
Product Version	
*Summary	
*Description	
Steps To Reproduce	
Additional Information	
Upload File (Maximum size: 2,097k)	Parcourir... Aucun fichier sélectionné.
View Status	public private
Report Stay	check to report more issues
* required	Submit Report

Рис. 6.4. Форма для отчета об ошибке

программного обеспечения, содержащегося в Kali. New Tool Requests (Запрос нового инструмента) используется для предложения новых средств упаковки и интеграции в дистрибутив Kali.

- Reproducibility (Воспроизводимость). В этом раскрывающемся списке указывается, возникает ли проблема предсказуемо или же это происходит случайным образом.
- Severity (Серьезность) и Priority (Приоритет). Эти параметры лучше оставить без изменений, поскольку они в основном предназначены для разработчиков. Последние используют данные параметры для сортировки списка проблем в соответствии с серьезностью проблемы и приоритетом ее обработки.
- Product Version (Версия продукта). В этом раскрывающемся списке указывается версия Kali Linux, которую вы используете (или ближайшая к ней). Хорошенько



подумайте, прежде чем сообщать об ошибке в старой версии, которая больше не поддерживается.

- Summary (mandatory) (Аннотация (обязательно)). Это, по сути, название вашего отчета об ошибке и первое, что увидят люди. Убедитесь в указании причины, по которой вы отправляете отчет. Избегайте общих описаний, таких как «X не работает», и вместо этого выбирайте «X с ошибкой Y при условии Z».
- Description (mandatory) (Описание (обязательно)). Это основной текст вашего отчета. В данном поле вы должны ввести всю информацию, собранную о проблеме. Не забывайте о рекомендациях, приведенных в предыдущем разделе.
- Steps to Reproduce (Действия по воспроизведению). В этом поле перечислите подробные инструкции, объясняющие, как запустить проблему.
- Additional Information (Дополнительная информация). В данном поле вы можете предоставить любую дополнительную информацию, которая, по вашему мнению, имеет отношение к проблеме. Если у вас есть идеи, как исправить или обойти проблему, то расскажите о них здесь.
- Upload File (Загрузить файл). Не все можно объяснить одним текстом. В этой строке вы можете прикрепить любые файлы: снимки экрана, чтобы показать ошибку, образцы документов, вызывающих проблему, файлы журналов и т. д.
- View Status (Статус отображения). Оставьте данный переключатель в положении public (открытый), чтобы все пользователи могли видеть ваш отчет об ошибке. Задействуйте положение private (закрытый) только для отчетов, связанных с безопасностью, и содержащих информацию о нераскрытых уязвимостях безопасности.

Подача отчета об ошибке в Debian

Debian использует (в основном) систему отслеживания ошибок на основе электронной почты, известную как Debbugs. Чтобы создать новый отчет об ошибке, нужно отправить электронное письмо (со специальным синтаксисом) на адрес submit@bugs.debian.org. После этого для вас будет выделен номер ошибки XXXXXX и вам сообщат, что вы можете отправить дополнительную информацию на адрес XXXXXX@bugs.debian.org. Каждая ошибка связана с пакетом Debian. Вы можете просмотреть все ошибки, относящиеся к данному пакету (включая ту, о которой хотите сообщить), на странице https://www.debian.org/Bugs/пакет. Вы можете увидеть историю данной ошибки на странице https://www.debian.org/Bugs/XXXXXX.

Настройка программы Reportbug. Несмотря на то что вы можете зарегистрировать новую ошибку с помощью простого электронного письма, мы рекомендуем использовать программу `reportbug`, призванную помочь составить основательный отчет об ошибке со всей необходимой информацией. В идеале вы должны запустить программу из системы Debian (например, на виртуальной машине, где воспроизвели проблему).

Первый запуск `reportbug` открывает сценарий конфигурации. Сначала выберите уровень навыка: Novice (Новичок) или Standard (Стандарт); мы используем последний, поскольку он обеспечивает более глубокое управление. Затем выберите интерфейс и введите личные данные. Наконец выберите пользовательский интерфейс. Сценарий конфигурации позволит применять локальный агент транспорта почты, SMTP-сервер или, в крайнем случае, SMTP-сервер Debian.

```
Welcome to reportbug! Since it looks like this is the first time you have
used reportbug, we are configuring its behavior. These settings will be
saved to the file "/root/.reportbugrc", which you will be free to edit
further.
Please choose the default operating mode for reportbug.

1 novice    Offer simple prompts, bypassing technical questions.

2 standard  Offer more extensive prompts, including asking about things
                  that a moderately sophisticated user would be expected to
                  know about Debian.

3 advanced  Like standard, but assumes you know a bit more about Debian,
                  including "incoming".

4 expert    Bypass most handholding measures and preliminary triage
                  routines. This mode should not be used by people unfamiliar
                  with Debian's policies and operating procedures.

Select mode: [novice] standard
Please choose the default interface for reportbug.

1 text   A text-oriented console user interface

2 gtk2   A graphical (GTK+) user interface.

3 urwid  A menu-based console user interface

Select interface: text
Will reportbug often have direct Internet access? (You should answer
yes to this question unless you know what you are doing and plan to
check whether duplicate reports have been filed via some other channel.)
[Y|n|q|?]? Y
What real name should be used for sending bug reports?
[root]> Raphaël Hertzog
Which of your email addresses should be used when sending bug reports?
(Note that this address will be visible in the bug tracking system, so you
may want to use a webmail address or another address with good spam
filtering capabilities.)
[root@localhost.localdomain]> buxy@kali.org
Do you have a "mail transport agent" (MTA) like Exim, Postfix or SSMTP
configured on this computer to send mail to the Internet? [y|N|q|?]? N
```

```
Please enter the name of your SMTP host. Usually it's called something
like "mail.example.org" or "smtp.example.org". If you need to use a
different port than default, use the <host>:<port> alternative
format. Just press ENTER if you don't have one or don't know, and so a
Debian SMTP host will be used.
>
Please enter the name of your proxy server. It should only use this
parameter if you are behind a firewall. The PROXY argument should be
formatted as a valid HTTP URL, including (if necessary) a port number; for
example, http://192.168.1.1:3128/. Just press ENTER if you don't have one
or don't know.
>
Default preferences file written. To reconfigure, re-run reportbug with
the "--configure" option.
```

Использование программы Reportbug. После завершения этапа настройки можно переходить непосредственно к созданию отчета об ошибке. Вам будет предложено указать имя пакета (хотя вы также можете указать его непосредственно в командной строке с помощью команды `reportbug` *пакет*).

```
Running 'reportbug' as root is probably insecure! Continue [y|N|q|?]? y
Please enter the name of the package in which you have found a problem, or
type 'other' to report a more general problem. If you don't know what
package the bug is in, please contact debian-user@lists.debian.org for
assistance.
> wireshark
```

Несмотря на рекомендации, приведенные выше, если вы не знаете, к какому пакету относится ошибка, то должны связаться с форумом поддержки Kali (см. раздел 6.2). На следующем этапе `reportbug` скачает список ошибок, выложенных для данного пакета, и позволит просмотреть их, на случай если вы сможете найти среди них свою.

```
*** Welcome to reportbug. Use ? for help at prompts. ***
Note: bug reports are publicly archived (including the email address of
the submitter).
Detected character set: UTF-8
Please change your locale if this is incorrect.

Using '"Raphaël Hertzog" <buxy@kali.org>' as your from address.
Getting status for wireshark...
Verifying package integrity...
Checking for newer versions at madison...
Will send report to Debian (per lsb_release).
Querying Debian BTS for reports on wireshark (source)...
35 bug reports found:

Bugs with severity important
   1) #478200 tshark: seems to ignore read filters when writing to…
```

```
   2) #776206 mergecap: Fails to create output file > 2GB
   3) #780089 wireshark: "On gnome wireshark has not title bar. Does…
Bugs with severity normal
   4) #151017 ethereal: "Protocol Hierarchy Statistics" give misleading…
   5) #275839 doesn't correctly dissect ESMTP pipelining
[...]
  35) #815122 wireshark: add OID 1.3.6.1.4.1.11129.2.4.2
(24-35/35) Is the bug you found listed above [y|N|b|m|r|q|s|f|e|?]? ?
y – Problem already reported; optionally add extra information.
N – (default) Problem not listed above; possibly check more.
b – Open the complete bugs list in a web browser.
m – Get more information about a bug (you can also enter a number
    without selecting "m" first).
r – Redisplay the last bugs shown.
q – I'm bored; quit please.
s – Skip remaining problems; file a new report immediately.
f – Filter bug list using a pattern.
e – Open the report using an e-mail client.
? – Display this help.
(24-35/35) Is the bug you found listed above [y|N|b|m|r|q|s|f|e|?]? n
Maintainer for wireshark is 'Balint Reczey <balint@balintreczey.hu>'.
Looking up dependencies of wireshark...
```

Если вы обнаружили, что ваша ошибка уже зарегистрирована, то можете отправить дополнительную информацию о ней. В противном случае вам предложат подать новый отчет об ошибке:

```
Briefly describe the problem (max. 100 characters allowed). This will be
the bug email subject, so keep the summary as concise as possible, for
example: "fails to send email" or "does not start with -q option
specified" (enter Ctrl+c to exit reportbug without reporting a bug).
> does not dissect protocol foobar
Rewriting subject to 'wireshark: does not dissect protocol foobar'
```

После изложения краткой аннотации вашей проблемы вы должны оценить ее серьезность в широком масштабе:

```
How would you rate the severity of this problem or report?

1 critical        makes unrelated software on the system (or the whole
                  system) break, or causes serious data loss, or
                  introduces a security hole on systems where you install
                  the package.
2 grave           makes the package in question unusable by most or all
                  users, or causes data loss, or introduces a security
                  hole allowing access to the accounts of users who use
                  the package.
3 serious         is a severe violation of Debian policy (that is, the
                  problem is a violation of a 'must' or 'required'
                  directive); may or may not affect the usability of the
```

```
                  package. Note that non-severe policy violations may be
                  'normal,' 'minor,' or 'wishlist' bugs. (Package
                  maintainers may also designate other bugs as 'serious'
                  and thus release-critical; however, end users should
                  not do so.). For the canonical list of issues worthing
                  a serious severity you can refer to this webpage:
                  http://release.debian.org/testing/rc_policy.txt
4 important       a bug which has a major effect on the usability of a
                  package, without rendering it completely unusable to
                  everyone.
5 does-not-build  a bug that stops the package from being built from source.
                  (This is a 'virtual severity'.)
6 normal          a bug that does not undermine the usability of the whole
                  package; for example, a problem with a particular
                  option or menu item.
7 minor           things like spelling mistakes and other minor cosmetic
                  errors that do not affect the core functionality of the
                  package.
8 wishlist        suggestions and requests for new features.

Please select a severity level: [normal]
```

Если вы не уверены в ответе, то просто оставьте уровень серьезности по умолчанию — нормальный (`normal`).

Вы также можете пометить свой отчет несколькими ключевыми словами:

```
Do any of the following apply to this report?

1 d-i            This bug is relevant to the development of debian-installer.
2 ipv6           This bug affects support for Internet Protocol version 6.
3 l10n           This bug reports a localization/internationalization issue.
4 lfs            This bug affects support for large files (over 2 gigabytes).
5 newcomer       This bug has a known solution but the maintainer requests
                 someone else implement it.
6 patch          You are including a patch to fix this problem.
7 upstream       This bug applies to the upstream part of the package.
8 none

Please select tags: (one at a time) [none]
```

Большинство тегов довольно непонятные, но если ваш отчет содержит решение проблемы, то вы должны выбрать тег `patch`.

Когда этот этап завершен, программа `reportbug` откроет текстовый редактор с шаблоном, который вы должны отредактировать (пример 6.2). Он содержит несколько вопросов, которые вам нужно удалить и вместо них ввести ответы, а также некую информацию о вашей системе, собранную автоматически. Обратите внимание на структуру первых нескольких строк. Их нельзя изменять, поскольку они будут проанализированы системой отслеживания ошибок, чтобы назначить отчет правильному пакету.

Пример 6.2. Шаблон, сгенерированный программой reportbug

```
Subject: wireshark: does not dissect protocol foobar

Package: wireshark
Version: 2.0.2+ga16e22e-1
Severity: normal

Dear Maintainer,

*** Reporter, please consider answering these questions, where appropriate ***

      * What led up to the situation?
      * What exactly did you do (or not do) that was effective (or
        ineffective)?
      * What was the outcome of this action?
      * What outcome did you expect instead?

*** End of the template - remove these template lines ***

-- System Information:
Debian Release: stretch/sid
  APT prefers testing
  APT policy: (500, 'testing')
Architecture: amd64 (x86_64)
Foreign Architectures: i386

Kernel: Linux 4.4.0-1-amd64 (SMP w/4 CPU cores)
Locale: LANG=fr_FR.utf8, LC_CTYPE=fr_FR.utf8 (charmap=UTF-8)
Shell: /bin/sh linked to /bin/dash
Init: systemd (via /run/systemd/system)

Versions of packages wireshark depends on:
ii wireshark-qt 2.0.2+ga16e22e-1

wireshark recommends no packages.

wireshark suggests no packages.

-- no debconf information
```

Сразу после сохранения отчета и закрытия текстового редактора вы вернетесь к программе `reportbug`, которая предоставит множество прочих опций и предложений касательно отправки итогового отчета.

```
Spawning sensible-editor...
Report will be sent to "Debian Bug Tracking System" <submit@bugs.debian.org>
Submit this report on wireshark (e to edit) [Y|n|a|c|e|i|l|m|p|q|d|t|s|?]? ?
Y - (default) Submit the bug report via email.
n - Don't submit the bug report; instead, save it in a temporary file (exits
reportbug).
```

```
a - Attach a file.
c - Change editor and re-edit.
e - Re-edit the bug report.
i - Include a text file.
l - Pipe the message through the pager.
m - Choose a mailer to edit the report.
p - print message to stdout.
q - Save it in a temporary file and quit.
d - Detach an attachment file.
t - Add tags.
s - Add a X-Debbugs-CC recipient (a CC but after BTS processing).
? - Display this help.
Submit this report on wireshark (e to edit) [Y|n|a|c|e|i|l|m|p|q|d|t|s|?]? Y
Saving a backup of the report at /tmp/reportbug-wireshark-backup-20160328-
19073-87oJWJ
Connecting to reportbug.debian.org via SMTP...

Bug report submitted to: "Debian Bug Tracking System" <submit@bugs.debian.org>
Copies will be sent after processing to:
  buxy@kali.org

If you want to provide additional information, please wait to receive the
bug tracking number via email; you may then send any extra information to
n@bugs.debian.org (e.g. 999999@bugs.debian.org), where n is the bug
number. Normally you will receive an acknowledgement via email including
the bug report number within an hour; if you haven't received a
confirmation, then the bug reporting process failed at some point
(reportbug or MTA failure, BTS maintenance, etc.).
```

Подача отчета об ошибке в другом проекте свободного программного обеспечения

Существует большое разнообразие проектов свободного программного обеспечения, использующих различные рабочие процессы и инструменты. Это разнообразие также касается и применяемых систем отслеживания ошибок. Хотя многие проекты размещены на веб-сервисе GitHub и используют систему GitHub Issues для отслеживания своих ошибок, есть и много других проектов, которые размещают собственные системы, основанные на Bugzilla, Trac, Redmine, Flyspray и др. Большинство из них сетевые и требуют регистрации для подачи нового отчета.

Мы не будем описывать здесь все системы отслеживания для других проектов свободного программного обеспечения. При желании вы можете самостоятельно изучить их особенности, но поскольку веб-сервис GitHub относительно популярен, то кратко рассмотрим его здесь. Как и в случае с другими системами отслеживания, вы должны сначала создать учетную запись и войти в нее. Затем перейдите на вкладку Issues (Проблемы) (рис. 6.5).

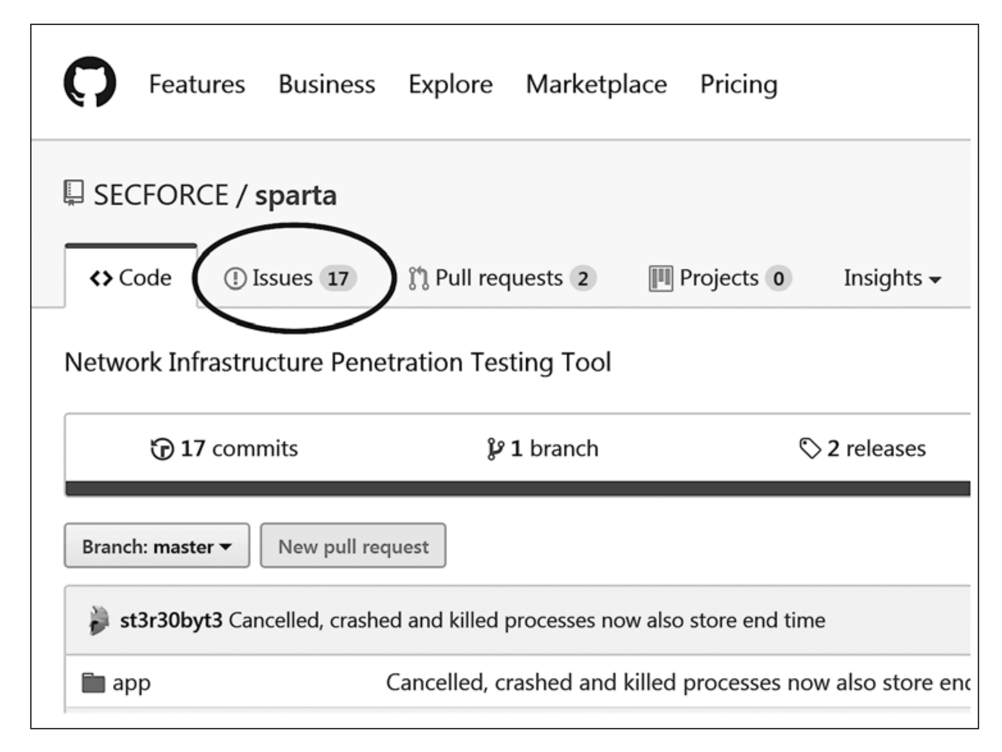


Рис. 6.5. Главная страница проекта GitHub

Затем вы можете просмотреть список зарегистрированных ошибок и выполнить среди них поиск. Если вы уверены, что ваша ошибка еще не зарегистрирована, то нажмите кнопку New issue (Новая проблема) (рис. 6.6).

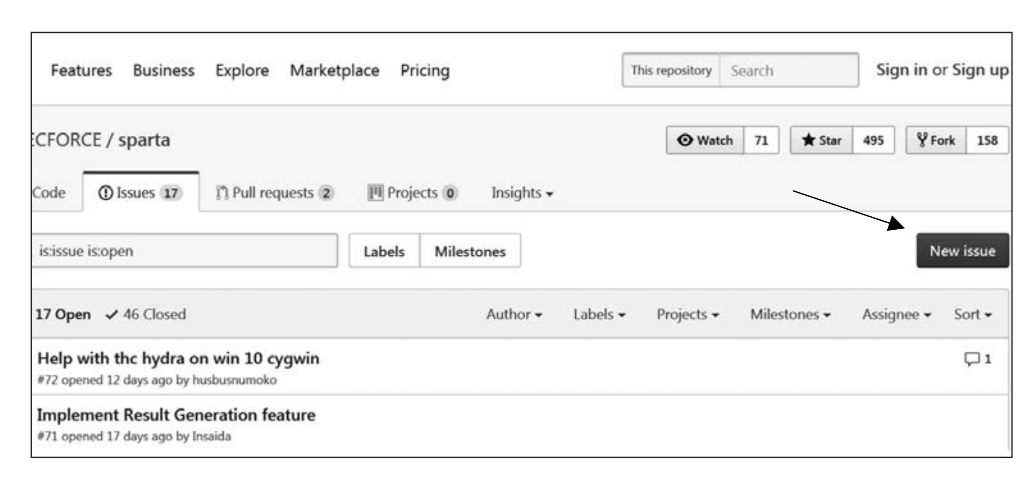


Рис. 6.6. Страница Issues проекта GitHub

Вы перешли на страницу, где необходимо описать свою проблему (рис. 6.7). Несмотря на отсутствие шаблона, подобного тому, что мы встретили в `reportbug`, механизм отчетности об ошибках довольно прост и позволяет прикреплять файлы, применять форматирование к тексту и др. Конечно, для достижения наилучших результатов обязательно следуйте нашим рекомендациям по созданию подробного и всеохватывающего отчета.

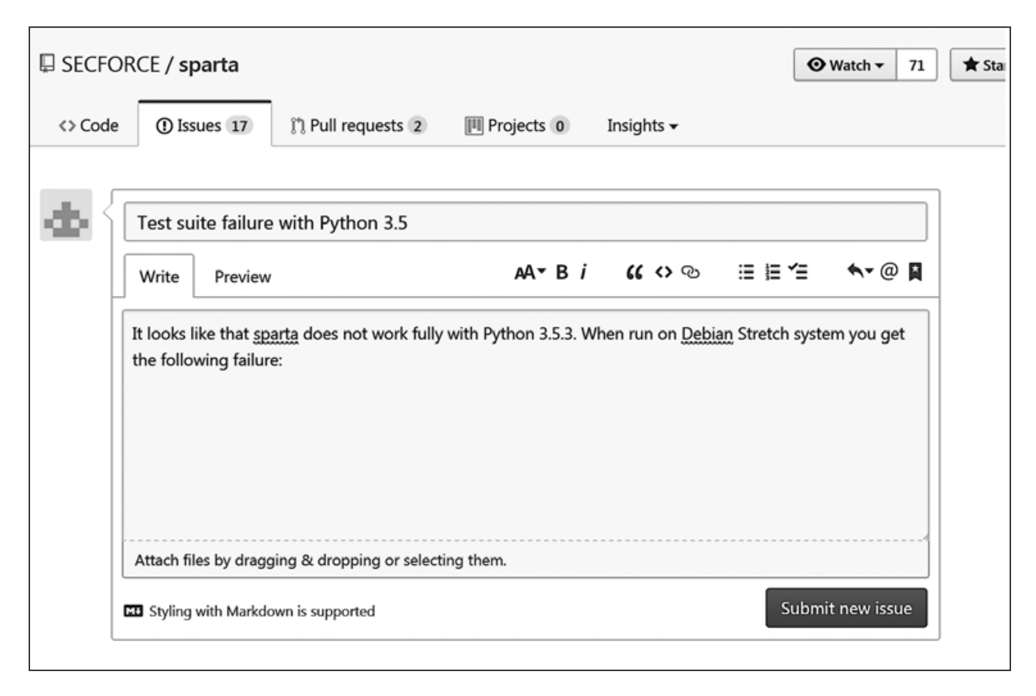


Рис. 6.7. Форма GitHub для регистрации новой проблемы

6.4. Резюме

В этой главе мы обсудили различные методы поиска документации и информации о программах, выяснили, как получить помощь в решении проблем, с которыми вы можете столкнуться. Рассмотрели руководства и страницы формата `info`, а также команды `apropos` и `info`. Представили системы отслеживания ошибок, дали несколько советов о том, как искать и отправлять правильные отчеты об ошибках, а также подсказали способ понять, кто владеет конкретной программой или проектом.

- Прежде чем вы сможете понять, что же на самом деле происходит, когда возникает проблема, вам необходимо знать теоретическую роль каждой программы, связанной с проблемой. Один из лучших способов сделать это — ознакомиться с документацией программы.

- Чтобы ознакомиться с руководством, достаточно ввести команду `man` *руководство*, указав имя команды после номера раздела (не обязательно).
- Команда `apropos` возвращает список руководств, в кратком описании которых упоминаются запрошенные ключевые слова, а также однострочное описание каждого из этих руководств.
- Проект GNU предоставил руководства для большинства своих программ в формате `info`. Поэтому многие руководства ссылаются на соответствующую документацию формата `info`.
- Каждый пакет содержит сопутствующую документацию, и даже к наименее документированным программам обычно прилагается файл `README`, хранящий интересную и/или важную информацию. Эта документация размещена в каталоге `/usr/share/doc/`*пакет*`/`.
- В большинстве случаев в архивах часто задаваемых вопросов или почтовых рассылок на официальном сайте программы можно найти описание проблемы, с которой вы столкнулись.
- Проект Kali хранит сборник полезной документации по адресу https://docs.kali.org/.
- Проект Kali Linux использует канал `#kali-linux` в IRC-сети Freenode. Вы можете задействовать ресурс chat.freenode.net в качестве IRC-сервера, используя порт 6667 для TLS-соединения или 6666 для соединения с открытым текстом. Чтобы присоединиться к обсуждениям в IRC, вы должны применить IRC-клиент `hexchat` (в графическом режиме) или `irssi` (в консольном режиме). Существует также веб-клиент, доступный на сайте http://webchat.freenode.net/.
- Официальные форумы сообщества для проекта Kali Linux расположены по адресу https://forums.kali.org/.
- Если вы обнаружите ошибку в программе, то можете найти соответствующий отчет об ошибке или подать собственный. Обязательно соблюдайте изложенные нами рекомендации, чтобы убедиться, что ваш отчет понятный и исчерпывающий, и увеличить шансы на своевременное устранение ошибки разработчиками.
- Некоторые отчеты об ошибках нужно отправить в Kali, а другие можно подать в Debian. Команда типа `dpkg -s` *пакет* `| grep ^Version:` выведет на экран номер версии с пометкой `kali`, если это пакет, модифицированный Kali.
- Идентифицировать проект более высокого уровня и найти, куда подавать отчет об ошибке, обычно легко. Вам просто нужно просмотреть сайт, ссылка на который содержится в метаданных пакета в поле `Homepage`.
- Kali использует сетевую систему отслеживания ошибок, расположенную на сайте https://bugs.kali.org/my_view_page.php, где вы можете просматривать отчеты об ошибках анонимно, но если хотите оставить комментарий или подать новый отчет об ошибке, то придется создать новую учетную запись.
- Debian задействует (в основном) систему отслеживания ошибок на основе электронной почты, известную как Debbugs. Чтобы создать новый отчет об ошибке, нужно отправить электронное письмо (со специальным синтаксисом)

на адрес submit@bugs.debian.org или воспользоваться программой `reportbug`, которая проведет через весь процесс подачи отчета.

- Хотя многие проекты размещены на веб-сервисе GitHub и используют систему GitHub Issues для отслеживания своих ошибок, есть также много других проектов, которые размещают собственные системы отслеживания. Возможно, вам придется ознакомиться с основами прочих систем, если понадобится регистрировать ошибки.

Теперь, когда у вас есть основные инструменты для навигации по Linux, установки и настройки Kali, а также для решения проблем системы и получения помощи, пришло время взглянуть на блокировку Kali, чтобы вы смогли защитить свою установку и данные своих клиентов.

ГЛАВА

7

Защита и контроль Kali Linux



Ключевые темы:

- политика безопасности;
- брандмауэр;
- команда iptables;
- мониторинг;
- протоколирование.

Как только вы начнете использовать Kali Linux для более конфиденциальных и высокопрофильных задач, вам, скорее всего, придется серьезнее отнестись к безопасности вашей установки. В этой главе мы вначале обсудим политику безопасности, выделив наиболее важные моменты при ее определении, и обратим внимание на некоторые угрозы для вашей системы и для вас как профессионала по безопасности. Мы также обсудим меры безопасности для ноутбуков и настольных систем и рассмотрим отдельно брандмауэры и фильтрацию пакетов. В завершение затронем инструменты и стратегии мониторинга и покажем наиболее эффективные способы их использования для обнаружения потенциальных угроз для вашей системы.

7.1. Определение политики безопасности

Нецелесообразно обсуждать безопасность в общих чертах, поскольку это понятие представляет собой широкий спектр концепций, инструментов и процедур, которые не являются универсальными. Выбор среди них требует точного представления ваших целей. Защита системы начинается с ответов на несколько вопросов. Поспешное и безрассудное внедрение произвольного набора утилит ведет к риску ошибочного определения аспектов безопасности.

Лучше всего изначально определить конкретную цель. Правильным подходом к решению этой задачи будут ответы на следующие вопросы.

- *Что* вы пытаетесь защитить? Политика безопасности будет отличаться в зависимости от того, что вы хотите защитить: компьютеры или данные. В последнем случае вам также нужно знать, какая именно информация требует защиты.
- *От чего* вы пытаетесь защититься? От утечки конфиденциальных данных? От случайной потери информации? От убытка, вызванного сбоем в предоставлении услуг?
- *От кого* вы пытаетесь защититься? Меры безопасности будут совершенно разными для защиты от опечатки простого пользователя системы и защиты от определенной группы злоумышленников.

Термин «риск» обычно используется для общего определения указанных факторов: что нужно защитить, что следует предотвратить и по чьей вине это может произойти. Для моделирования риска требуются ответы на все три вопроса. Основываясь на полученной модели, можно разработать политику безопасности и реализовать ее с помощью конкретных действий.

Неизменный вопрос

Брюс Шнайер, мировой эксперт по вопросам безопасности (не только компьютерной), пытается противостоять одному из основных мифов безопасности, действуя под девизом: «Безопасность — это процесс, а не продукт». Активы, которые нуждаются в защите, со временем меняются, так же как и угрозы и средства, доступные потенциальным злоумышленникам. Даже если политика безопасности изначально была идеально разработана и реализована, вы никогда не должны останавливаться на достигнутом. Компоненты риска развиваются, и методы его предотвращения должны развиваться соответственно.

Кроме того, следует учитывать дополнительные сдерживающие факторы, которые могут ограничивать диапазон доступных политик. На что вы готовы пойти ради защиты системы? Этот вопрос имеет большое значение для выбора политики. Очень часто ответ определяется только с точки зрения денежных издержек, но следует учитывать и другие элементы, такие как возможные неудобства, которым подвергнутся пользователи системы, или ухудшение ее производительности.

Как только риск смоделирован, вы можете задумываться о разработке подходящей политики безопасности.

Существуют крайности, которые стоит рассматривать при принятии решения об уровне необходимой безопасности. С одной стороны, чрезвычайно просто обеспечить базовую безопасность системы.

Например, если система, определенная для защиты, включает только подержанный компьютер, который используется лишь для сложения пары чисел в конце дня, то вполне разумным решением было бы не предпринимать ничего особенного для ее защиты. Истинная ценность такой системы низкая, а ценность данных и вовсе равна нулю, так как они не хранятся на компьютере. Потенциальный злоумышленник, проникший в эту систему, получит только калькулятор. Стоимость защиты подобной системы, вероятно, будет больше, чем убытки от взлома.

Противоположной ситуацией станет случай защиты конфиденциальности секретных данных самым полным способом с преодолением любых ограничений. В этом случае подходящим решением будет полное уничтожение информации (безопасное стирание файлов, измельчение жестких дисков на мелкие кусочки, затем растворение этих кусочков в кислоте и т. д.). Если существует дополнительное требование о том, что данные должны храниться для дальнейшего использования (не обязательно в постоянной доступности) и стоимость по-прежнему не является сдерживающим фактором, то лучшей идеей будет хранение данных на пластинах из сплава иридия и платины в бомбоубежищах под горами по всему миру, каждое из которых (разумеется) засекречено и охраняется целой армией.

Хотя эти методы могут показаться преувеличенными, тем не менее они могут быть подходящими решениями для определенных рисков, поскольку позволяют достичь намеченных целей при заданных ограничениях. Исходя из обоснованного решения, никакая политика безопасности не является более или менее достаточной, чем любая другая.

Возвращаясь к более типичному случаю, информационная система может быть сегментирована в совместимые и преимущественно независимые подсистемы. Все они имеют собственные требования и ограничения, поэтому оценку риска и разработку политики безопасности следует проводить отдельно для каждой из таких подсистем. Нужно всегда помнить о том, что малую поверхность атаки легче защитить, чем большую. Сетевые организации должны быть спроектированы так: уязвимые сервисы необходимо сконцентрировать на небольшом количестве компьютеров, и последние должны быть доступны через минимальное количество маршрутов или контрольных точек. Логика проста: легче защитить контрольные точки, чем все уязвимые компьютеры от всего внешнего мира. Именно в этот момент

становится очевидной польза сетевой фильтрации (в том числе брандмауэрами). Данную фильтрацию можно реализовать с помощью специального оборудования, но более простым и гибким решением является использование программного брандмауэра, подобного тому, что интегрирован в ядро Linux.

7.2. Возможные меры безопасности

Как было сказано выше, нет единого ответа на вопрос о том, как защитить Kali Linux. Все зависит от способа его использования и от того, что именно вы пытаетесь защитить.

На сервере

Если вы используете Kali Linux на общедоступном сервере, то стоит защитить сетевые сервисы, изменив все пароли по умолчанию, которые могут быть настроены, и, вероятно, путем ограничения доступа к ним с помощью брандмауэра (разделы 7.3 «Защита сетевых сервисов» и 7.4 «Брандмауэр или фильтрация пакетов» соответственно, см. ниже).

Если вы передаете данные учетных записей пользователей непосредственно на сервере либо на одном из сетевых сервисов, убедитесь, что установили надежные пароли (они должны выдерживать атаки полным перебором). В то же время можно настроить программу `fail2ban`, которая значительно усложняет взлом паролей полным перебором по сети (отфильтровывая IP-адреса, превышающие лимит неудачных попыток входа в систему). Установить `fail2ban` можно с помощью команды `apt update` и затем `apt install fail2ban`.

Если вы используете веб-сервисы, настраивайте их работу через протокол HTTPS, чтобы сетевые посредники не отслеживали ваш трафик (который может включать файлы аутентификации cookie).

На ноутбуке

Ноутбук специалиста по тестированию на проникновение не претерпевает тех же рисков, что и открытый сервер: например, вы менее подвержены случайным атакам взломщика-дилетанта, а если это и произойдет, у вас, вероятно, не будет активных сетевых сервисов в этот момент.

Реальный риск часто возникает, когда вы едете от одного клиента к другому. Например, ваш ноутбук может быть украден во время поездки или изъят таможенниками. Вот почему стоит использовать полное шифрование диска (см. подраздел «Установка на полностью зашифрованную файловую систему» раздела 4.2) и, возможно, также настроить функцию `nuke` (см. врезку «Установка пароля самоуничтожения для дополнительной безопасности» в главе 9): данные, которые вы собрали во время вашей работы, являются конфиденциальными и требуют максимальной защиты.

Вам также могут потребоваться правила брандмауэра (см. ниже раздел 7.4), но не для той же цели, что и на сервере. Вероятно, вы захотите запретить весь исходящий трафик, кроме трафика, генерируемого вашим VPN-доступом. Эти настройки подобны настройкам безопасности сети, так что когда VPN перестанет работать, вы сразу же заметите это (вместо того, чтобы возвращаться к локальному сетевому доступу). Таким образом, вы не выдаете IP-адреса своих клиентов при просмотре веб-страниц или других сетевых действиях. Кроме того, если вы выполняете локальное внутреннее взаимодействие, то лучше всего непрестанно контролировать свою деятельность, чтобы уменьшить шум, создаваемый в сети, который может привлечь внимание клиентов и их системы защиты.

7.3. Защита сетевых сервисов

Рекомендуется отключить сервисы, которые вы не используете. Kali упрощает эту задачу, поскольку большинство сетевых сервисов по умолчанию уже отключены.

Пока сервисы остаются отключенными, они не представляют угрозы безопасности. Однако вы должны быть осторожны при их включении ввиду следующих факторов.

- По умолчанию у них нет брандмауэра, поэтому, если они прослушивают все сетевые интерфейсы, то в значительной степени доступны для общественности.
- Некоторые сервисы не имеют учетных данных и позволяют устанавливать их при первом использовании; другие имеют стандартные (и, следовательно, широко известные) учетные данные. Убедитесь, что вы (пере)установили пароль, который известен только вам.
- Многие сервисы выпускаются с правами `root` (с полными правами администратора), поэтому последствия несанкционированного доступа или нарушения безопасности обычно являются серьезными.

Учетные данные по умолчанию

Мы не будем перечислять здесь все инструменты, которые поставляются с учетными данными по умолчанию. Вместо этого вы должны проверить файл README.Debian для соответствующих пакетов, а также страницы https://docs.kali.org/ и https://tools.kali.org/ с целью узнать, нуждается ли сервис в специальном обслуживании, чтобы обеспечить необходимую безопасность.

Если вы запускаетесь в режиме реального времени, то паролем учетной записи root является toor. Таким образом, вы не должны включать SSH перед сменой пароля учетной записи root или прежде чем настроить в конфигурации учетной записи запрет входа на основе пароля.

Обратите внимание также на известный факт, что проект BeEF (из уже установленного пакета beef-xss) имеет учетные данные по умолчанию: имя пользователя beef и пароль beef, которые заданы «принудительно» в файле конфигурации.

7.4. Брандмауэр или фильтрация пакетов

Брандмауэр — это часть компьютерного оборудования с аппаратным обеспечением, программным обеспечением или и тем и другим, которая анализирует входящие или исходящие сетевые пакеты (приходящие или исходящие из локальной сети) и пропускает только те, которые соответствуют определенным предварительно заданным условиям.

Фильтрующий сетевой шлюз — тип брандмауэра, который защищает всю сеть. Обычно он устанавливается на специально выделенный компьютер, сконфигурированный как шлюз для сети, таким образом, что может анализировать все входящие и исходящие из сети пакеты. В качестве альтернативы существует локальный брандмауэр, представляющий собой сервис ПО, работающий на одном определенном компьютере, чтобы фильтровать или ограничивать доступ к ряду сервисов на этом компьютере или, возможно, предотвращать исходящие соединения от шпионского программного обеспечения, которое пользователь мог установить случайно или умышленно.

В ядро Linux встроен брандмауэр `netfilter`. Не существует единого решения для настройки любого брандмауэра, так как требования сети и пользователя разнятся. Тем не менее вы можете контролировать `netfilter` из пользовательского пространства с помощью команд `iptables` и `ip6tables`. Разница между последними заключается в том, что первая работает для сетей IPv4, тогда как вторая функционирует на IPv6. Поскольку оба стека сетевых протоколов, вероятно, будут работать в течение многих лет, то оба инструмента должны использоваться параллельно. Вы также можете применять отличную утилиту `fwbuilder` на основе графического интерфейса, который обеспечивает графическое представление правил фильтрации.

Однако если вы решили настроить `netfilter` (реализация брандмауэра Linux), то рассмотрим подробнее, как он работает.

Поведение сетевого фильтра Netfilter

Фильтр `Netfilter` использует четыре различные таблицы, в которых хранятся правила, регламентирующие три вида операций над пакетами:

- `filter` касается правил фильтрации (принятие, отказ или игнорирование пакета);
- `nat` (Network Address Translation — трансляция сетевых адресов) касается трансляции исходных или целевых адресов и портов пакетов;
- `mangle` относится к другим изменениям в IP-пакетах (включая поле ToS (Type of Service — тип сервиса) и опции);
- `raw` допускает другие изменения в пакетах, проведенные вручную, до того как они (пакеты) достигнут системы отслеживания соединения.

Каждая таблица содержит списки правил, называемые *цепями*. Брандмауэр использует стандартные цепи для обработки пакетов на основе предопределенных условий. Администратор может создавать другие цепи, которые будут применяться только при передаче одной из стандартных цепей (прямо или косвенно).

Таблица `filter` содержит три стандартные цепи:

- `INPUT` — касается пакетов, целью которых является сам брандмауэр;
- `OUTPUT` — относится к пакетам, исходящим от брандмауэра;
- `FORWARD` — относится к пакетам, проходящим через брандмауэр (который не является ни их источником, ни местом назначения).

Таблица `nat` также имеет три стандартные цепи:

- `PREROUTING` — для изменения пакетов сразу после их поступления;
- `POSTROUTING` — для изменения пакетов, когда они готовы к отправке;
- `OUTPUT` — для изменения пакетов, сгенерированных самим брандмауэром.

Эти цепи изображены на рис. 7.1.

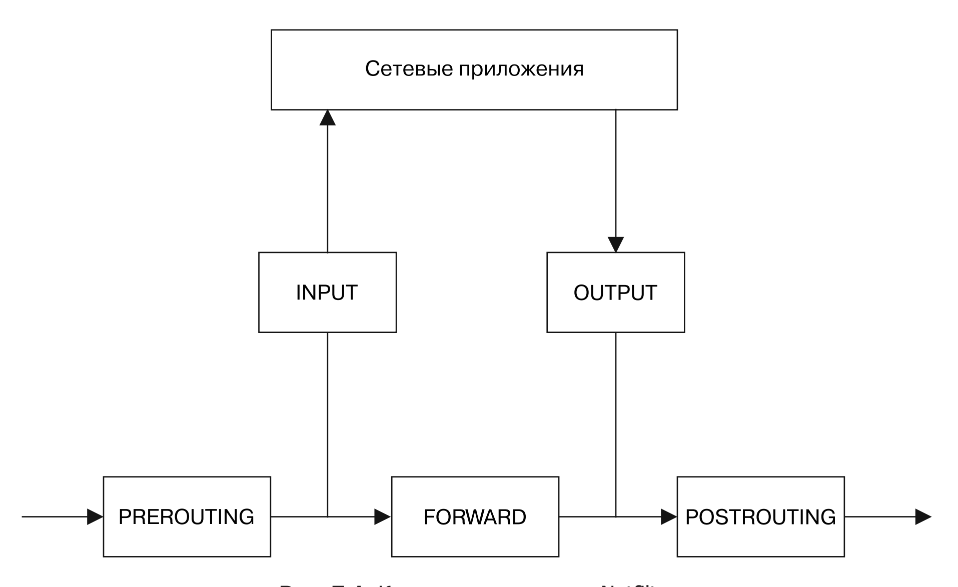


Рис. 7.1. Как вызываются цепи Netfilter

Каждая цепь представляет собой список правил; каждое правило есть набор условий и действие, выполняемое при выполнении условий. При обработке пакета брандмауэр сканирует соответствующую цепь, одно правило за другим, и когда условия для одного правила выполняются, перескакивает (jump) (отсюда параметр `-j` в командах) к указанному действию для продолжения обработки. Наиболее распространенные типы поведения стандартизированы, и для них существуют специальные действия. Выполнение одного из этих стандартных

действий прерывает обработку цепочки, поскольку дальнейшая судьба пакетов уже предрешена (не принимая во внимание исключение, упомянутое ниже). Далее перечислены действия Netfilter.

- ACCEPT (ПРИНЯТЬ) — разрешить пакету двигаться далее по своему маршруту.
- REJECT (ОТКЛОНИТЬ) — отклонить пакет с помощью пакета ошибок ICMP (Internet control message protocol — протокол межсетевых управляющих сообщений) (параметр --reject-with *тип* для iptables определяет тип ошибки для отклонения).
- DROP (СБРОСИТЬ) — удалить (игнорировать) пакет.
- LOG (ЗАРЕГИСТРИРОВАТЬ) — зарегистрировать (через демон syslogd) сообщение с описанием пакета. Обратите внимание, что это действие не прерывает обработку, а выполнение цепи продолжается со следующего правила, поэтому регистрация отклоненных пакетов требует как правила LOG, так и REJECT/DROP. Общие параметры, связанные с регистрацией, включают:
  - --log-level, с предупреждением по умолчанию, указывает уровень серьезности syslog;
  - --log-prefix позволяет указать префикс текста, чтобы различать зарегистрированные сообщения;
  - --log-tcp-sequence, --log-tcp-options и --log-ip-options обозначают дополнительные данные, которые должны быть помещены в сообщение: порядковый номер TCP, параметры TCP и параметры IP соответственно.
- ULOG — зарегистрировать сообщение через ulogd, который может быть лучше адаптирован и более эффективен, чем syslogd для обработки большого количества сообщений; обратите внимание, что это действие, подобно LOG, также возвращает обработку к следующему правилу в вызывающей цепи.
- имя_цепи — перейти к указанной цепи и оценить ее правила.
- RETURN (ВЕРНУТЬ) — прервать обработку текущей цепи и вернуться к вызывающей цепочке; если текущая цепочка является стандартной, то вызывающей цепочки нет, поэтому вместо нее выполняется действие по умолчанию (определенное с помощью параметра -P для iptables).
- SNAT (только в таблице nat) — применить источник трансляции сетевых адресов (Source Network Address Translation, SNAT). Дополнительные параметры описывают точные изменения, которые нужно применить, включая параметр --to-source *адрес:порт*, который определяет новый источник IP-адреса и/или порта.
- DNAT (только в таблице nat) — применить назначение трансляции сетевых адресов (Destination Network Address Translation, DNAT). Дополнительные параметры описывают точные изменения, которые нужно использовать, включая параметр --to-destination *адрес:порт*, который определяет новый источник IP-адреса и/или порта.
- MASQUERADE (МАСКИРОВКА) (только в таблице nat) — применить *маскировку* (особый случай Source NAT).

- `REDIRECT` (ПЕРЕНАПРАВЛЕНИЕ) (только в таблице `nat`) — открыто перенаправить пакет в данный порт самого брандмауэра. Можно использовать для настройки открытого сервера веб-прокси, который работает без конфигурации на стороне клиента, и в то время, когда клиент считает, что подключается к получателю, фактически сообщения проходят через прокси-сервер. Параметр `--to-ports` *порт(-ы)* указывает порт или диапазон портов, куда должны быть перенаправлены пакеты.

Другие действия, особенно те, которые касаются таблицы `mangle`, не вошли в данный подраздел. Их полный список вы найдете на страницах руководств `iptables (8)` и `ip6tables (8)`.

Что такое ICMP? Протокол межсетевых управляющих сообщений (ICMP) — протокол, используемый для передачи вспомогательной информации с помощью сообщений. Он проверяет сетевое соединение, применяя команду ping, отправляющую ICMP-сообщение запроса отклика, на которое получатель должен отвечать соответствующим ICMP-сообщением (откликом). Протокол сигнализирует, что брандмауэр отклонил пакет, указывает на переполнение в буфере приема, предлагает лучший маршрут для следующих пакетов в соединении и т. д. Определяется несколькими документами типа RFC. В числе первых документов можно назвать RFC777 (http://www.faqs.org/rfcs/rfc777.html) и RFC792 (http://www.faqs.org/rfcs/rfc792.html), но затем появились и другие, которые расширили и/или внесли изменения в протокол.

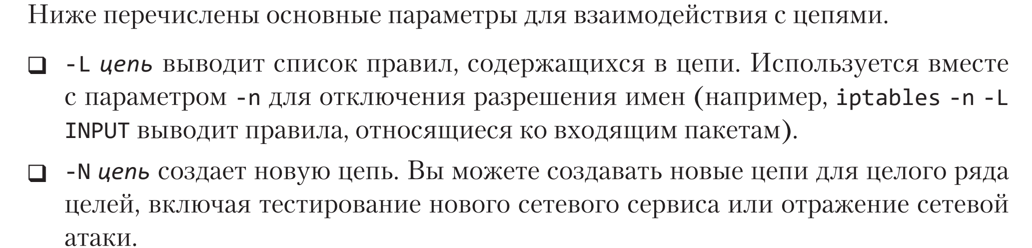
Для справки: буфер приема представляет собой небольшую зону памяти, в которой данные хранятся в период между моментом, когда они поступают из сети, и моментом, когда ядро их обрабатывает. Если эта зона заполнена, то новые данные не могут быть получены и ICMP сигнализирует о проблеме, чтобы эмиттер мог замедлить скорость передачи (вследствие чего в идеале через некоторое время должно быть достигнуто равновесие).

Обратите внимание: хотя сеть IPv4 может работать и без ICMP, для сети IPv6 строго требуется протокол ICMPv6, поскольку он объединяет несколько функций, которые для IPv4 распространяются через ICMPv4, протокол членства в группах Интернета (Internet Group Membership Protocol, IGMP) и протокол определения адреса (Address Resolution Protocol, ARP). Протокол ICMPv6 определен в документе RFC4443 (http://www.faqs.org/rfcs/rfc4443.html).

Синтаксис команд iptables и ip6tables

Команды `iptables` и `ip6tables` используются для управления таблицами, цепями и правилами. Их параметр `-t table` указывает, с какой таблицей работать (по умолчанию таблица `filter`).

Команды



Ниже перечислены основные параметры для взаимодействия с цепями.

- `-L` *цепь* выводит список правил, содержащихся в цепи. Используется вместе с параметром `-n` для отключения разрешения имен (например, `iptables -n -L INPUT` выводит правила, относящиеся ко входящим пакетам).
- `-N` *цепь* создает новую цепь. Вы можете создавать новые цепи для целого ряда целей, включая тестирование нового сетевого сервиса или отражение сетевой атаки.
- `-X` *цепь* удаляет пустую и неиспользуемую цепь (например, `iptables -X ddos-attack`).
- `-A` *цепь правило* добавляет правило в конце заданной цепи. Помните, правила обрабатываются сверху вниз, не забывайте учитывать этот момент при добавлении правил.
- `-I` *цепь номер_правила правило* вставляет правило перед правилом с указанным номером. Как и в параметре `-A`, учитывайте порядок обработки при вводе новых правил в цепь.
- `-D` *цепь номер_правила* (или `-D` *цепь правило*) удаляет правило в цепи; первый синтаксис указывает, что правило под определенным номером должно быть удалено (команда `iptables -L --line-numbers` выводит на экран номера правил), а второй идентифицирует правило к удалению по его сути.
- `-F` *цепь* сбрасывает цепь (удаляет все ее правила). Например, чтобы удалить все правила, связанные с исходящими пакетами, вы должны ввести команду `iptables -F OUTPUT.` Если ни одна цепь не указана, то удаляются все правила в таблице.
- `-P` *цепь действие* определяет действие по умолчанию или «политику» для данной цепи. Обратите внимание: такая политика присуща только для стандартных цепей. Чтобы удалить весь входящий трафик по умолчанию, вы должны выполнить команду `iptables -P INPUT DROP`.

Правила

Каждое правило задается в соответствии со следующим синтаксисом: *условия* `-j` *действие параметры_действия*. Если в одном правиле описано несколько условий, то критерием является объединение (логическое И) условий, которое обладает ограничением, по меньшей мере таким же, как и каждое отдельно взятое условие.

Условие `-p` *протокол* соответствует полю протокола IP-пакета. Наиболее распространенными значениями являются `tcp`, `udp`, `icmp` и `icmpv6`. Это условие можно дополнить условиями касательно TCP-портов с помощью параметров `--source-port` *порт* и `--destination-port` *порт*.

Отрицательные условия Добавление восклицательного знака перед условием означает его отрицание. Например, отрицание условия по параметру -p означает «любой пакет с протоколом, отличным от того, который указан». Этот механизм отрицания можно применять к любым другим условиям.

Условие `-s` *адрес* или `-s` *сеть/маска* соответствует исходному (`source`) адресу пакета. Соответственно, `-d` *адрес* или `-d` *сеть/маска* соответствует адресу назначения (`destination`).

Условие `-i` *интерфейс* выбирает пакеты, исходящие из заданного сетевого интерфейса; `-o` *интерфейс* — пакеты, выходящие на определенный интерфейс.

Условие `--state` *статус* соответствует статусу пакета в соединении (для этого требуется модуль ядра `ipt_ conntrack` для отслеживания соединения). Статус `NEW` описывает пакет, запускающий новое соединение, статус `ESTABLISHED` соответствует пакетам, принадлежащим к уже существующему соединению, и статус `RELATED` соответствует пакетам, инициирующим новое соединение, связанное с существующим (что полезно для соединений `ftp`-данных в активном режиме протокола FTP).

Существует множество доступных параметров для `iptables` и `ip6tables`, и для их освоения потребуется немало времени. Однако один из них, который вы будете использовать чаще всего, — это параметр, блокирующий вредоносный сетевой трафик с хоста или диапазона хостов. Например, чтобы незаметно блокировать входящий трафик с IP-адреса 10.0.1.5 и 31.13.74.0/24 класса С подсети, нужно выполнить ряд команд:

```
# iptables -A INPUT -s 10.0.1.5 -j DROP
# iptables -A INPUT -s 31.13.74.0/24 -j DROP
# iptables -n -L INPUT
Chain INPUT (policy ACCEPT)
target     prot opt source             destination
DROP       all  --  10.0.1.5           0.0.0.0/0
DROP       all  --  31.13.74.0/24      0.0.0.0/0
```

Другая команда `iptables` часто применяется с целью разрешения сетевого трафика для определенного сервиса или порта. Чтобы пользователи могли подключаться к SSH, HTTP и IMAP, вы должны выполнить следующие команды:

```
# iptables -A INPUT -m state --state NEW -p tcp --dport 22 -j ACCEPT
# iptables -A INPUT -m state --state NEW -p tcp --dport 80 -j ACCEPT
# iptables -A INPUT -m state --state NEW -p tcp --dport 143 -j ACCEPT
# iptables -n -L INPUT
Chain INPUT (policy ACCEPT)
target     prot opt source             destination
DROP       all  --  10.0.1.5           0.0.0.0/0
DROP       all  --  31.13.74.0/24      0.0.0.0/0
ACCEPT     tcp  --  0.0.0.0/0          0.0.0.0/0          state NEW tcp dpt:22
ACCEPT     tcp  --  0.0.0.0/0          0.0.0.0/0          state NEW tcp dpt:80
ACCEPT     tcp  --  0.0.0.0/0          0.0.0.0/0          state NEW tcp dpt:143
```

Правилом хорошей компьютерной *гигиены* является очистка старых и ненужных правил. Самый простой способ удалить правило iptables — ссылаться на правила по номеру строки, которые вы можете получить с помощью параметра --line-numbers. Будьте внимательны: при сбросе правила все последующие правила в цепочке будут перенумерованы.

```
# iptables -n -L INPUT --line-numbers
Chain INPUT (policy ACCEPT)
num  target     prot  opt  source          destination
1    DROP       all   --   10.0.1.5        0.0.0.0/0
2    DROP       all   --   31.13.74.0/24   0.0.0.0/0
3    ACCEPT     tcp   --   0.0.0.0/0       0.0.0.0/0        state NEW tcp dpt:22
4    ACCEPT     tcp   --   0.0.0.0/0       0.0.0.0/0        state NEW tcp dpt:80
5    ACCEPT     tcp   --   0.0.0.0/0       0.0.0.0/0        state NEW tcp dpt:143
# iptables -D INPUT 2
# iptables -D INPUT 1
# iptables -n -L INPUT --line-numbers
Chain INPUT (policy ACCEPT)
num  target     prot  opt  source          destination
1    ACCEPT     tcp   --   0.0.0.0/0       0.0.0.0/0        state NEW tcp dpt:22
2    ACCEPT     tcp   --   0.0.0.0/0       0.0.0.0/0        state NEW tcp dpt:80
3    ACCEPT     tcp   --   0.0.0.0/0       0.0.0.0/0        state NEW tcp dpt:143
```

Существуют более специфические условия, зависящие от общих условий, описанных выше. Для получения дополнительной информации обратитесь к руководствам iptables (8) и ip6tables (8).

Создание правил

Для каждого нового правила требуется один вызов iptables или ip6tables. Ввод этих команд вручную может быть утомительным, так что вызовы обычно хранятся в сценарии, и, как следствие, система автоматически настраивается одинаково при каждой загрузке компьютера. Данный сценарий можно написать вручную, но вам может быть также интересно подготовить его с помощью высокоуровневого инструмента, такого как fwbuilder.

```
# apt install fwbuilder
```

Принцип прост. На первом этапе опишите все элементы, которые будут задействованы в новых правилах:

- сам брандмауэр с его сетевыми интерфейсами;
- сети с соответствующими диапазонами IP-адресов;
- серверы;
- порты, принадлежащие службам, размещенным на серверах.

Затем создайте правила с помощью простых действий перетаскивания объектов, как показано на рис. 7.2. Несколько контекстных меню могут изменить условие (например, отрицать его). Затем нужно выбрать и настроить действие.

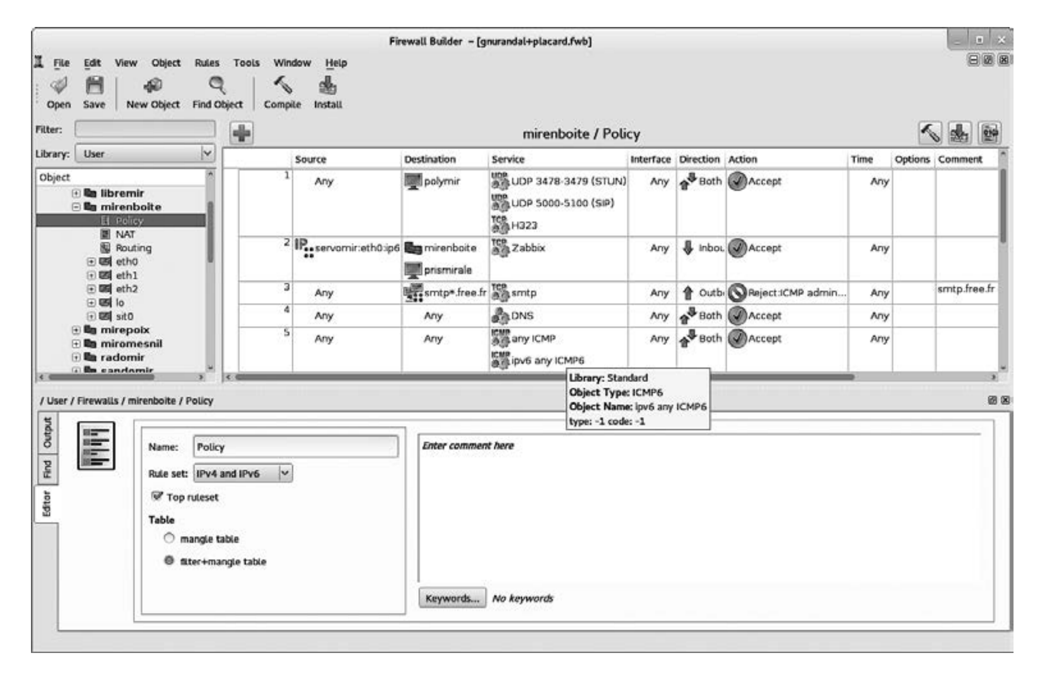


Рис.7.2. Главное окно fwbuilder

Что касается IPv6, то вы можете либо создать два разных набора правил для IPv4 и IPv6, либо создать только одно и позволить `fwbuilder` преобразовывать правила в соответствии с адресами, назначенными объектам.

Инструмент `fwbuilder` создаст сценарий, настраивающий брандмауэр в соответствии с правилами, которые вы определили. Его модульная архитектура позволяет генерировать сценарии, предназначенные для разных систем, включая `iptables` для Linux, `ipf` для FreeBSD и `pf` для OpenBSD.

Установка правил при каждой загрузке

Чтобы внедрять правила брандмауэра при каждой загрузке машины, вам необходимо зарегистрировать сценарий конфигурации в директиве `up` файла `/etc/network/interfaces`. В следующем примере сценарий хранится в `/usr/local/etc/arrakis.fw`.

```
auto eth0
iface eth0 inet static
    address 192.168.0.1
    network 192.168.0.0
    netmask 255.255.255.0
    broadcast 192.168.0.255
    up /usr/local/etc/arrakis.fw
```

В этом примере предполагается, что вы используете пакет `ifupdown` для настройки сетевых интерфейсов. Если вы применяете что-то другое (скажем, `NetworkManager` или `systemd-networkd`), то обратитесь к соответствующей документации, чтобы узнать, как выполнить сценарий после запуска интерфейса.

7.5. Мониторинг и протоколирование

Конфиденциальность и защита данных — важные аспекты безопасности, но не менее важно обеспечить доступность сервисов. В качестве администратора и специалиста по безопасности вы должны следить за тем, чтобы все работало должным образом, и ваша ответственность — своевременно выявлять аномальное поведение и ухудшения в работе сервисов. Программное обеспечение для мониторинга и протоколирования играет ключевую роль в этом аспекте безопасности, обеспечивая понимание того, что происходит в системе и в сети.

В этом разделе мы рассмотрим ряд инструментов, которые можно использовать для мониторинга нескольких аспектов системы Kali.

Мониторинг журналов с помощью программы logcheck

Программа `logcheck` отслеживает файлы журнала каждый час по умолчанию и отправляет нестандартные сообщения журнала в сообщения электронной почты администратору для их дальнейшего анализа.

Список контролируемых файлов хранится по адресу `/etc/logcheck/logcheck.logfiles`. Значения по умолчанию будут работать должным образом, если файл `/etc/rsyslog.conf` не был полностью перестроен.

Программа `logcheck` может отчитываться, используя разные уровни детализации: `paranoid` (параноидальный), `server` (серверный) и `workstation` (для рабочих станций). Режим `paranoid` очень многословен и, вероятно, должен быть ограничен конкретными серверами, такими как брандмауэры. Режим `server` используется по умолчанию и рекомендуется для большинства серверов. Режим `workstation`, очевидно, предназначен для рабочих станций и чрезвычайно сжат, отфильтровывая больше сообщений, чем другие «собратья».

Во всех трех случаях `logcheck`, вероятно, должен быть настроен для исключения дополнительных сообщений (в зависимости от установленных сервисов), если вы не хотите получать ежечасные партии длинных незарегистрированных электронных писем. Поскольку механизм выбора сообщений довольно сложный, то файл `/usr/share/doc/logcheck-database/README.logcheck-database.gz` обязателен к прочтению при возникновении сложностей.

Применяемые правила можно разделить на несколько типов:

- те, которые квалифицируют сообщение как попытку взлома (хранятся в файле в каталоге `/etc/logcheck/cracking.d/`);

- проигнорированные попытки взлома (`/etc/logcheck/cracking.ignore.d/`);
- те, которые классифицируют сообщение как предупреждение системы безопасности (`/etc/logcheck/violations.d/`);
- проигнорированные предупреждения системы безопасности (`/etc/logcheck/violations.ignore.d/`);
- наконец те, которые применяются к остальным сообщениям (рассматриваются как *системные события*).

Файлы `ignore.d` используются (очевидно) для игнорирования сообщений. Например, сообщение, помеченное как попытка взлома или предупреждение системы безопасности (по правилу, хранящееся в файле `/etc/logcheck/violations.d/myfile`), может быть проигнорировано только правилом в файле `/etc/logcheck/violations.ignore.d/myfile` или в файле `/etc/logcheck/changes.ignore.d/myfile-`*расширение*.

О системном событии всегда сообщается, если только правило в одном из каталогов `/etc/logcheck/ignore.d.{paranoid, server, workstation}/` не указывает, что это событие следует игнорировать. Разумеется, учитываются лишь те каталоги, чьи уровни детализации равняются выбранному режиму работы или превышают его.

Мониторинг активности в режиме реального времени

Интерактивный инструмент `top` отображает список процессов, запущенных в данный момент. Сортировка по умолчанию основана на текущей загруженности процессора и может быть получена с помощью ключа `P`. Другие сортировки указаний содержат сортировку по занятой памяти (ключ `M`), общему времени процессора (ключ `T`) и идентификатору процесса (ключ `N`). Ключ `k` завершает процесс с введенным идентификатором. Ключ `r` изменяет приоритет процесса.

Когда система кажется перегруженной, `top` — отличный инструмент, позволяющий увидеть, какие процессы конкурируют за процессорное время или потребляют слишком много памяти. Так, часто бывает интересно проверить, соответствуют ли процессы, потребляющие ресурсы, реальным сервисам, которые должны быть размещены на компьютере. Неизвестный процесс, работающий как www-data, должен действительно выделяться из списка, и его следует изучить, поскольку он, вероятнее всего, является примером программного обеспечения, установленного и выполняемого в системе с помощью уязвимости в веб-приложении.

Инструмент `top` очень гибкий, и руководство к нему содержит подробную информацию о том, как настроить его интерфейс и адаптировать к вашим личным потребностям и привычкам.

Графический инструмент `gnome-system-monitor` похож на `top` и обеспечивает выполнение примерно тех же функций.

Обнаружение изменений

После установки и настройки системы большинство ее файлов должны оставаться относительно неизменными до тех пор, пока система не будет обновлена. Поэтому рекомендуется следить за изменениями в этих файлах, так как любое непредвиденное изменение может быть причиной тревоги, и его нужно исследовать. В этом подразделе представлены некоторые из наиболее распространенных инструментов, используемых для мониторинга системных файлов, обнаружения изменений и, возможно, уведомления вас как администратора системы.

Проверка пакетов с помощью инструмента dpkg --verify

Инструмент `dpkg --verify` (или `dpkg -V`) весьма интересен, поскольку отображает системные файлы, которые были изменены (предположительно злоумышленником), но этот вывод следует воспринимать с долей скепсиса.

При выполнении своей работы `dpkg` полагается на контрольные суммы, хранящиеся в его собственной базе данных, которые, в свою очередь, хранятся на жестком диске (могут быть найдены в `/var/lib/dpkg/info/package.md5sums`). Зная это, внимательный злоумышленник будет изменять системные файлы, чтобы они содержали новые контрольные суммы для поврежденных файлов, или более продвинутый злоумышленник может взломать пакет в вашем зеркале Debian. Для защиты от подобной атаки используйте систему проверки цифровой подписи APT (см. подраздел «Проверка подлинности пакета» раздела 8.3) для правильной проверки пакетов.

Что такое контрольная сумма файла?

Напомним: контрольная сумма — это значение, часто число (хотя и в шестнадцатеричной системе исчисления), которое содержит своего рода подпись для содержимого файла. Данная подпись рассчитывается с помощью алгоритма (например, известные MD5 или SHA1), сей факт более или менее гарантирует, что даже самые незначительные изменения в содержимом приведут к изменению контрольной суммы; это известно как «эффект лавины». Простая цифровая подпись служит эффективным средством для проверки того, было ли изменено содержимое файла. Эти алгоритмы необратимы; другими словами, для большинства из них знание контрольной суммы не позволяет узнать соответствующее содержимое. Недавние математические достижения, по-видимому, ослабили абсолютность этих принципов, но их использование до сих пор не ставится под сомнение, поскольку создание другого содержания, дающего одну и ту же контрольную сумму, по-прежнему представляется довольно сложной задачей.

Запуск `dpkg -V` проверяет все установленные пакеты и выводит на экран строку для каждого файла, который не прошел проверку. Каждый символ обозначает проверку на конкретные метаданные. К сожалению, `dpkg` не хранит метаданные, необходимые для большинства тестов и, таким образом, выводит вместо них вопросительные знаки. В настоящее время только в случае провала проверки контрольной суммы на третьей позиции будет указана цифра 5.

```
# dpkg -V
??5??????   /lib/systemd/system/ssh.service
??5?????? c /etc/libvirt/qemu/networks/default.xml
??5?????? c /etc/lvm/lvm.conf
??5?????? c /etc/salt/roster
```

В приведенном выше примере `dpkg` сообщает об изменении файла сервиса SSH, которое администратор произвел в пакетированном файле вместо использования соответствующей замены `/etc/systemd/system/ssh.service` (который будет храниться под каталогом `/etc`, как и должны храниться любые изменения конфигурации). В примере также перечислено несколько файлов конфигурации (обозначенных буквой `c` во втором поле), измененных по правилам.

Мониторинг файлов: AIDE. Инструмент `AIDE` (Advanced Intrusion Detection Environment — современная среда обнаружения вторжений) проверяет целостность файла и обнаруживает любые изменения, которые не соответствуют ранее записанному образу действительной системы. Образ хранится в виде базы данных (`/var/lib/aide/aide.db`), содержащей соответствующую информацию обо всех файлах системы (контрольные суммы, разрешения, временные метки и т. д.).

Вы можете установить `AIDE`, выполнив команду `apt update`, а затем `apt install aide`. Сначала вы должны инициализировать базу данных с помощью команды `aideinit`; она будет запускаться ежедневно (через сценарий `/etc/cron.daily/aide`), чтобы проверить наличие существенных изменений. Если они обнаружены, то `AIDE` записывает их в файлы журнала (`/var/log/aide/*.log`) и отправляет свои результаты администратору по электронной почте.

> **Защита базы данных** Поскольку AIDE использует локальную базу данных для сравнения состояний файлов, то достоверность результатов напрямую связана с достоверностью базы. Если злоумышленник получает права root для взломанной системы, то сможет заменить базу данных и замести следы взлома. Один из способов предотвращения подобной деятельности — хранение справочных данных на носителях, предназначенных только для чтения.

Вы можете редактировать файл `/etc/default/aide`, чтобы настроить поведение пакета *aide*. Конфигурация самого инструмента AIDE хранится в файлах `/etc/aide/aide.conf` и `/etc/aide/aide.conf.d/` (обычно эти файлы используются только

командой update-aide.conf для генерации файла /var/lib/aide/aide.conf.autogenerated). Конфигурация указывает, какие свойства нужно проверить. Например, содержимое файлов журналов изменяется в обычном режиме, и такие преобразования можно игнорировать, если разрешения этих файлов остаются незатронутыми, однако как содержимое, так и разрешения исполняемых программ должны быть неизменными. Надо признать, что синтаксис конфигурации не полностью интуитивно понятен, и мы рекомендуем дополнительно ознакомиться с руководством aide.conf (5).

Новая версия базы данных создается ежедневно в /var/lib/aide/aide.db.new; если все зарегистрированные изменения были правомерными, то базу данных можно заменить.

Инструмент Tripwire очень похож на AIDE; даже синтаксис файла конфигурации почти одинаковый. Основное дополнение, предоставляемое tripwire, — это механизм подписи файла конфигурации, который не позволяет злоумышленнику указать на другую версию справочной базы данных.

Инструмент Samhain предлагает аналогичные функции, а также некоторые дополнительные функции, помогающие обнаружить руткиты (см. врезку «Пакеты checksecurity и chkrootkit/rkhunter» ниже). Он также может быть развернут глобально во всей сети и записывать результаты своей работы на центральном сервере (с подписью).

Пакеты checksecurity и chkrootkit/rkhunter

Пакет checksecurity состоит из нескольких небольших сценариев, которые выполняют основные проверки в системе (поиск пустых паролей, новых файлов setuid и т. д.) и предупреждают об обнаружении этих условий. Несмотря на многообещающее имя пакета, вы не должны полагаться исключительно на него с целью убедиться, что система Linux в безопасности.

Пакеты chkrootkit и rkhunter обнаруживают определенные руткиты, потенциально установленные в системе. Напомним, что это части программного обеспечения, предназначенные для скрытия взлома системы, притом позволяющие незаметно сохранять контроль над машиной. Тесты надежны не на 100 %, но обычно их результаты могут привлечь ваше внимание к потенциальным проблемам.

7.6. Резюме

В этой главе мы рассмотрели понятие политики безопасности, подчеркнув различные моменты, которые следует учитывать при определении такой политики, и обратив внимание на ряд возможных угроз вашей системе и лично вам как специалисту по безопасности. Мы обсудили меры безопасности для ноутбуков и настольных систем, а также брандмауэры и фильтрацию пакетов.

Наконец, мы представили инструменты и стратегии мониторинга и показали, как наилучшим образом реализовать их в целях обнаружения потенциальных угроз для вашей системы.

- Найдите время для определения всеохватывающей политики безопасности.
- Если вы используете Kali на общедоступном сервере, то измените все пароли по умолчанию для сервисов, которые могут быть настроены, и ограничьте их доступ с помощью брандмауэра (см. разделы 7.3 и 7.4 соответственно) перед их запуском.
- Используйте программу `fail2ban` для обнаружения и блокировки атак с угадыванием пароля и взлома пароля методом полного перебора.
- Если вы запускаете веб-сервисы, то размещайте их на HTTPS, чтобы сетевые посредники не отслеживали ваш трафик (который может включать файлы аутентификации cookie).
- Существенный риск часто возникает, когда вы едете от одного клиента к другому. Например, ваш ноутбук может быть украден во время поездки или изъят на таможне. Будьте готовы к подобным неприятностям, используя полное шифрование диска (см. подраздел «Установка на полностью зашифрованную файловую систему» раздела 4.2), а также рассмотрите функцию `nuke` (см. врезку «Установка пароля самоуничтожения для дополнительной безопасности» в главе 9) для защиты данных ваших клиентов.
- Установите правила брандмауэра (см. раздел 7.4), чтобы запретить весь исходящий трафик, кроме трафика, генерируемого вашим VPN-доступом. Это подобно безопасности сети, так что, когда VPN перестанет работать, вы сразу же заметите это (вместо того чтобы возвращаться к локальному сетевому доступу).
- Отключить сервисы, которые вы не используете. Kali упрощает эту задачу, поскольку большинство сетевых сервисов по умолчанию уже отключены.
- В ядро Linux встроен брандмауэр `netfilter`. Не существует единого решения для настройки любого брандмауэра, так как требования сети и пользователя разнятся. Тем не менее вы можете контролировать `netfilter` из пользовательского пространства с помощью команд `iptables` и `ip6tables`.
- Программа `logcheck` отслеживает файлы журнала каждый час по умолчанию и отправляет нестандартные сообщения журнала в сообщения электронной почты администратору для их дальнейшего анализа.
- Интерактивный инструмент `top` отображает список процессов, запущенных в данный момент.
- Инструмент `dpkg --verify` (или `dpkg -V`) отображает измененные (предположительно злоумышленником) системные файлы, но полагается на контрольные суммы, которые умный взломщик может изменить.
- Инструмент `AIDE` проверяет целостность файла и обнаруживает любые изменения, которые не соответствуют ранее записанному образу действительной системы.

- Инструмент Tripwire очень похож на AIDE, но использует механизм подписи файла конфигурации, который не позволяет злоумышленнику указать на другую версию справочной базы данных.
- Рассмотрите возможность использования пакетов rkhunter, checkecurity и chkrootkit для обнаружения руткитов в вашей системе.

В следующей главе мы рассмотрим основы Debian и управление пакетами. Вы быстро осознаете всю мощь, лежащую в основе Debian Kali, и узнаете, как разработчики ее используют. Обратите внимание: глава 8 довольно насыщенная, однако очень важно, чтобы вы понимали суть Debian и механизмы управления пакетами, если планируете стать уверенным пользователем Kali.

ГЛАВА

8

Управление пакетами Debian



Ключевые темы:

- dpkg;
- APT;
- sources.list;
- обновления;
- репозитории пакетов.

После того как вы ознакомились с основами Linux, пришло время изучить систему управления пакетами дистрибутива на базе Debian. В дистрибутивах, подобных Kali, пакет Debian представляет собой канонический способ сделать программное обеспечение доступным для конечных пользователей. Понимание системы управления пакетами даст представление о том, каким образом структурирована Kali, позволит более эффективно решать проблемы и быстро находить помощь и документацию для широкого спектра инструментов и утилит, включенных в Kali Linux.

В данной главе мы представим вашему вниманию систему управления пакетами Debian и познакомим с `dpkg` и набором инструментов APT. Одно из основных преимуществ Kali Linux — гибкость системы управления пакетами, которая с помощью этих инструментов обеспечивает практически непрерывную установку, обновление, удаление и обработку прикладного ПО и даже самой базовой операционной системы. Очень важно понять, как работает данная система, чтобы максимально задействовать Kali и оптимизировать ваши усилия. Дни болезненных компиляций, провальных обновлений, проблемы отладки `gcc`, `make` и `configure` давно прошли, но количество доступных приложений значительно выросло, и сейчас вам нужно понимать, как работают средства, созданные для их использования. Этот навык также необходим, поскольку существует огромное количество инструментов безопасности, которые по причине лицензирования или из-за других нюансов не могут быть включены в Kali, но представлены пакетами Debian для скачивания. Очень важно понимать, каким образом обрабатывать и устанавливать эти пакеты и как они влияют на систему, особенно в тех случаях, когда ситуация складывается вопреки ожиданиям.

Мы начнем с базового обзора APT, опишем структуру и содержимое двоичных и исходных пакетов, посмотрим на некоторые базовые утилиты и сценарии и затем копнем чуть глубже, чтобы помочь вам выжать максимум из этой эффективной пакетной системы и набора инструментов.

8.1. Введение в APT

Для затравки представим некоторые базовые определения, общий обзор и небольшую историю пакетов Debian, начиная наше повествование с инструментов `dpkg` и APT.

Взаимосвязь между APT и dpkg

Пакет Debian представляет собой сжатый архив программного приложения. *Бинарный пакет* (binary package (файл с расширением `.deb`)) содержит файлы, которые могут быть использованы напрямую (такие как программы или документация), в то время как *исходный пакет* (source package) включает исходный код для программного обеспечения, а также инструкции, необходимые для создания бинарных пакетов. В пакет Debian входят файлы приложения, а также другие *метаданные*,

в том числе названия зависимостей, которые нужны приложению, и сценарии, разрешающие выполнение команд на разных стадиях жизненного цикла пакета (установка, удаление и обновление).

Инструмент `dpkg` создан для обработки и установки пакетов с расширением `.deb`, но если он обнаруживает зависимость, которая не может быть удовлетворена (вроде отсутствующей библиотеки), то это помешает установке пакетов. В подобных случаях `dpkg` перечислит отсутствующие зависимости, так как у него просто нет вариантов действия или встроенной логики для обработки пакетов, которые должны удовлетворить эти зависимости. Инструмент APT (Advanced Package Tool), включая `apt` и `apt-get`, был разработан для устранения указанных недостатков и таким образом может автоматически решать подобные проблемы. В этой главе мы поговорим об инструментах `dpkg` и APT.

Базовой командой для обработки пакетов Debian в системе является `dpkg`, которая выполняет установку или анализ пакетов с расширением `.deb` и их содержимого. Тем не менее `dpkg` имеет доступ лишь к части данных о системе Debian: знает о том, что установлено в системе и что вы указываете в командной строке, однако не имеет представления о других доступных пакетах. Таким образом, этот инструмент не сможет правильно работать, если зависимость не будет выполнена. APT устраняет эти ограничения.

APT — набор инструментов, которые помогают управлять пакетами Debian или приложениями в вашей системе Debian. Вы можете использовать APT для установки и удаления приложений, обновления пакетов и даже обновления всей системы. Магия APT заключается в том, что он является полноценной системой управления пакетами, которая будет не только устанавливать или удалять пакеты, но и рассматривать требования и зависимости пакетированного приложения и пытаться удовлетворить их все автоматически. APT полагается на `dpkg`, однако, несмотря на это, отличается от него. APT устанавливает последнюю версию пакета из онлайн-источника и работает так, чтобы разрешить зависимости, в то время как `dpkg` устанавливает пакет, расположенный в вашей локальной системе, и не разрешает зависимости автоматически.

Если вы работаете в данной сфере достаточно долго, чтобы помнить о компиляции программ с помощью `gcc` (а также с помощью утилит, таких как `make` и `configure`), то, вероятно, помните, что это был довольно болезненный процесс, особенно когда приложение имело несколько зависимостей. Расшифровывая различные предупреждения и сообщения об ошибках, вы могли определить, какая часть кода была нерабочей, и чаще всего ошибка возникала из-за отсутствующей библиотеки или другой зависимости. Затем вы отслеживали эту недостающую библиотеку или зависимость, исправляли ее и повторяли попытку. Далее, если повезет, то компиляция завершалась, но часто процесс сборки снова обрывался, ссылаясь на другую нарушенную зависимость.

APT был разработан для того, чтобы помочь справиться с данной проблемой, сопоставить программные требования и зависимости и разрешить их. Это нестандартное решение для Kali Linux, но оно не является защищенным от неумелого обращения. Важно понимать, как работает система пакетирования Debian и Kali,

поскольку вам придется устанавливать пакеты, обновлять программное обеспечение или устранять проблемы, связанные с пакетами. Вы будете использовать APT в своей повседневной работе с Kali Linux, и в этой главе мы познакомим вас с APT и покажем, как устанавливать, удалять, обновлять пакеты и управлять ими, и даже продемонстрируем способы перемещения пакетов между разными дистрибутивами Linux. Мы также поговорим о графических инструментах, которые применяют APT, покажем, как проверять подлинность пакетов, и углубимся в концепцию дистрибутива со скользящим релизом — особой техники, ежедневно обновляющей вашу систему Kali.

Прежде чем мы покажем, как применять `dpkg` и APT для установки пакетов и управления ими, очень важно подробнее рассмотреть некоторые внутренние действия APT и обсудить терминологию, используемую в этом вопросе.

Источник пакета и исходный пакет Слово «исходный/источник (source)» может иметь два разных значения. Исходный пакет (source package) — это пакет, содержащий исходный код программы, и не следует путать его с источником пакета (package source) — хранилищем (сайтом, FTP-сервером, CD-ROM, локальным каталогом и т. д.), которое включает пакет.

APT извлекает свои пакеты из репозитория, хранилища пакетов или просто источника пакета. Файл `/etc/apt/sources.list` перечисляет различные хранилища (или источники), которые содержат пакеты Debian.

Подробности о файле sources.list

Файл `sources.list` — ключевой файл конфигурации для определения источника пакетов, и поэтому очень важно понимать, как он формируется и как его настраивать, поскольку APT не станет работать без правильно определенного списка источника пакетов. Обсудим синтаксис файла. Для начала взглянем на различные репозитории, которые используются Kali Linux, и обсудим зеркала и зеркальные перенаправления, и только после этого вы будете готовы применять APT.

Каждая активная строчка файла `/etc/apt/sources.list` (и файлов `/etc/apt/sources.list.d/*.list`) содержит описание источника, состоящее из трех частей, разделенных пробелами. Строки комментариев начинаются с символа #:

```
# deb cdrom:[Debian GNU/Linux 2016.1 _Kali-rolling_ - Official Snapshot amd64
  ➥LIVE/ INSTALL Binary 20160830-11:29]/ kali-rolling contrib main non-free

deb http://http.kali.org/kali kali-rolling main non-free contrib
```

Взглянем на синтаксис этого файла. Первое поле обозначает тип источника:

- `deb` для бинарных пакетов;
- `deb-src` для исходных пакетов.

Второе поле указывает на базовый URL источника: он может состоять из зеркала Debian или любого другого архива пакетов, настроенного третьей стороной. URL может начинаться с символов `file://` для локального источника, установленного в иерархии файлов системы, с `http://` для источника, доступного с веб-сервера, или с `ftp://` для источника, доступного на FTP-сервере. URL также может начинаться с символов `cdrom:` для установки с диска CD-ROM/DVD-ROM/Blu-ray, хотя этот способ используется реже, поскольку сетевые методы установки становятся все более популярными.

Запись `cdrom` описывает ваше устройство CD/DVD-ROM. В отличие от других записей, CD-ROM не всегда доступен, поскольку он должен быть вставлен в привод, и, как правило, за один раз можно считывать информацию только с одного диска. Из-за этого источники управляются несколько иначе и должны быть добавлены с помощью программы `apt-cdrom`, обычно запускаемой с параметром `add`. Затем появится запрос вставить диск в привод, где его содержимое будет просматриваться в поисках файлов `Packages`. Программа использует последние для обновления базы данных пакетов (эта операция обычно выполняется командой `apt update`). Затем APT запросит диск, если потребуется пакет, хранящийся на нем.

Синтаксис последнего поля зависит от структуры репозитория. В самых простых случаях вы можете указать подкаталог (с необходимым знаком слеша) нужного источника (это обычно обозначение `./`, указывающее на отсутствие подкаталога, — тогда пакеты находятся непосредственно по указанному URL). Но в большинстве случаев репозитории будут структурированы, подобно зеркалу Debian, со множеством дистрибутивов, каждый из которых обладает большим количеством компонентов. В таком случае укажите выбранный дистрибутив, а затем компоненты (или разделы), которые нужно включить. Уделим немного внимания этим разделам.

Debian и Kali используют три раздела, чтобы дифференцировать пакеты согласно лицензии, выбранной авторами каждого из проектов.

`Main` содержит все пакеты, которые соответствуют критериям Debian по определению свободного ПО (Debian Free Software Guidelines, https://www.debian.org/social_contract#guidelines).

Архивы типа `non-free` отличаются, так как содержат программное обеспечение, которое не (полностью) соответствует данным критериям, но тем не менее может распространяться без ограничения.

`Contrib` (contributions) представляет собой набор программ с открытым исходным кодом, которые не способны функционировать без отдельных элементов non-free. Последние могут включать программное обеспечение из раздела `non-free` или non-free-файлов, например игровые ПЗУ, BIOS консолей и т. д. `Contrib` также содержит бесплатное ПО, для компиляции которого требуются патентованные элементы, такие как VirtualBox, в свою очередь требующий non-free-компилятор для создания ряда своих файлов.

Теперь рассмотрим стандартные источники пакетов Kali Linux, или репозитории.

Репозитории Kali

Стандартный файл `sources.list` для системы, работающей на Kali Linux, относится к одному репозиторию (kali-rolling) и трем ранее упомянутым компонентам: `main`, `contrib` и `non-free`:

```
# Main Kali repository
deb http://http.kali.org/kali kali-rolling main contrib non-free
```

Рассмотрим различные репозитории Kali.

Репозиторий Kali Rolling

Это основной репозиторий для конечных пользователей. Он всегда должен содержать самые новые и устанавливаемые пакеты. Управляется инструментом, который объединяет Debian Testing и специализированные пакеты Kali, гарантируя таким образом, что зависимости каждого пакета могут быть удовлетворены в Kali Rolling. Другими словами, исключая вероятность любой ошибки в сценариях поддержки, все пакеты должны быть установлены.

Ввиду того что Debian Testing развивается ежедневно, эволюционирует и Kali Rolling. Специализированные пакеты Kali также регулярно обновляются, поскольку мы отслеживаем новейшие выпуски самых важных пакетов.

Репозиторий Kali-Dev

Не предназначен для общего пользования. Это пространство, в котором разработчики Kali решают проблемы зависимостей, возникающие в результате слияния специализированных пакетов Kali в Debian Testing.

Кроме того, это место, куда в первую очередь загружаются обновленные пакеты, поэтому, если вам нужно обновление, которое было выпущено недавно, но еще не добралось до kali-rolling, то можете получить его из данного репозитория. Не рекомендуется для обычных пользователей.

Репозиторий Kali-Bleeding-Edge

Содержит пакеты, автоматически созданные из соответствующего репозитория Git (или Subversion). Положительный момент — вы сразу же получаете доступ к последним функциям и исправлениям ошибок менее чем через 24 часа после того, как они были выпущены. Это идеальный способ проверить, исправлена ли ошибка, о которой вы сообщали ранее.

Недостатком является то, что эти пакеты не были протестированы или проверены: если внесенные изменения повлияли на пакетирование (добавив новую зависимость), то данный пакет может не работать. Поэтому репозиторий помечен таким образом, что APT не устанавливает пакеты из него автоматически, в частности во время обновления.

Вы можете зарегистрировать репозиторий, отредактировав файл `/etc/apt/sources.list` или создав новый файл в каталоге `/etc/apt/sources.list.d`, что является лучшим вариантом, поскольку оставляет исходный системный файл `sources.list` без изменений. В данном примере мы решили создать отдельный файл `/etc/apt/sources.list.d/kali-bleeding-edge.list` следующим образом:

```
# Kali Bleeding Edge repository
deb http://http.kali.org/kali kali-bleeding-edge main contrib non-free
```

Зеркала Kali Linux

Выдержки из файла `sources.list`, указанные выше, относятся к `http.kali.org`: это сервер, на котором работает MirrorBrain (http://mirrorbrain.org/), перенаправляющий ваши HTTP-запросы на официальное зеркало, находящееся рядом с вами. MirrorBrain контролирует каждое зеркало с целью гарантировать, что оно работает должным образом и обновлено; он всегда перенаправит вас на хорошее зеркало.

Отладка перенаправления зеркал

Если у вас возникает проблема с зеркалом (например, из-за неудачного выполнения команды apt update), вы можете использовать команду curl -sI, чтобы увидеть, куда именно вас перенаправили:

```
$ curl -sI http://http.kali.org/README
HTTP/1.1 302 Found
Date: Mon, 11 Apr 2016 09:43:21 GMT
Server: Apache/2.4.10 (Debian)
X-MirrorBrain-Mirror: ftp.free.fr
X-MirrorBrain-Realm: country
Link: <http://http.kali.org/README.meta4>; rel=describedby;
      ➥type="application/metalink4+xml"
Link: <http://ftp.free.fr/pub/kali/README>; rel=duplicate;
      ➥pri=1; geo=fr
Link: <http://de-rien.fr/kali/README>; rel=duplicate; pri=2;
      ➥geo=fr
Link: <http://ftp.halifax.rwth-aachen.de/kali/README>;
      ➥rel=duplicate; pri=3; geo=de
Link: <http://ftp.belnet.be/kali/kali/README>;
      ➥rel=duplicate; pri=4; geo=be
Link: <http://ftp2.nluug.nl/os/Linux/distr/kali/README>;
      ➥rel=duplicate; pri=5; geo=nl
Location: http://ftp.free.fr/pub/kali/README
Content-Type: text/html; charset=iso-8859-1
```

Если проблема не решилась, то можете отредактировать файл /etc/apt/sources.list и принудительно добавить имя другого известного рабочего зеркала вместо (или до) записи http.kali.org.

У нас также есть второй экземпляр MirrorBrain: где `http.kali.org` размещает репозитории пакетов, а `cdimage.kali.org` размещает выпущенные ISO-образы (http://cdimage.kali.org/).

Если вы хотите запросить список официальных зеркал Kali Linux, то можете добавить суффикс `.mirrorlist` в любой допустимый URL, указывающий на `http.kali.org` или `cdimage.kali.org` (http://http.kali.org/README.mirrorlist, http://cdimage.kali.org/README.mirrorlist).

Эти списки не являются полными из-за ряда ограничений MirrorBrain (в частности, зеркала, зависящие от некоторых стран, не отображаются в списке, если вы не находитесь в данной стране). Но они содержат лучшие зеркала: эти зеркала в хорошем состоянии и имеют большую пропускную способность.

8.2. Основное взаимодействие пакетов в Debian

Теперь, вооружившись базовым пониманием структуры APT, рассмотрим некоторые базовые взаимодействия пакетов, включая инициализацию APT; установку, удаление и очистку пакетов; модернизацию системы Kali Linux. Затем перейдем к командной строке, чтобы взглянуть на некоторые графические инструменты APT.

Инициализация APT

APT — обширный проект и набор утилит, в первоначальную планировку которого входил графический интерфейс. С точки зрения клиента, он сосредоточен вокруг инструмента командной строки `apt-get`, а также `apt`, разработанного позднее для устранения недостатков дизайна `apt-get`.

Существуют графические альтернативы, созданные другими программистами, в том числе `synaptic` и `aptitude`, которые мы обсудим немного позже. Мы предпочитаем использовать `apt`, как будет показано в последующих примерах. Мы также подробно объясним ряд основных различий синтаксиса между инструментами по мере их возникновения.

При работе с APT вы должны сначала скачать список доступных пакетов, используя команду `apt update`. В зависимости от скорости вашего подключения это может занять некоторое время, поскольку список различных пакетов, список источников и файлы трансляций выросли наряду с разработкой Debian. Конечно, установка с помощью CD/DVD происходит намного быстрее, так как источники являются локальными для вашего компьютера.

Установка пакетов

Благодаря продуманному дизайну системы пакетов Debian вы можете легко устанавливать пакеты с их зависимостями или без. Рассмотрим установку пакета с помощью инструментов `dpkg` и `apt`.

Установка пакетов с помощью dpkg

Инструмент dpkg — основной, его обычно используют (прямо или косвенно через APT) для установки пакета. Кроме того, это отличный вариант, если вы работаете в автономном режиме, поскольку инструмент не требует подключения к Интернету. Помните: dpkg не установит никаких зависимостей, которые могут потребоваться для пакета. Чтобы установить пакет с помощью dpkg, просто укажите параметр -i или --install и путь к файлу с расширением .deb. Подразумевается, что вы ранее скачали (или получили каким-то другим способом) файл .deb для устанавливаемого пакета.

```
# dpkg -i man-db_2.7.0.2-5_amd64.deb
(Reading database ... 86425 files and directories currently installed.)
Preparing to unpack man-db_2.7.0.2-5_amd64.deb ...
Unpacking man-db (2.7.0.2-5) over (2.7.0.2-4) ...
Setting up man-db (2.7.0.2-5) ...
Updating database of manual pages ...
Processing triggers for mime-support (3.58) ...
```

Мы можем видеть пошагово результат выполнения команды dpkg и соответственно заметить, в какой момент возникает вероятность какой-либо ошибки. Параметр -i или --install автоматически выполняет два этапа: распаковывает пакет и запускает сценарии конфигурации. Вы можете выполнить эти два шага самостоятельно (как это обычно делает apt) с помощью параметров --unpack и --configure соответственно:

```
# dpkg --unpack man-db_2.7.0.2-5_amd64.deb
(Reading database ... 86425 files and directories currently installed.)
Preparing to unpack man-db_2.7.0.2-5_amd64.deb ...
Unpacking man-db (2.7.0.2-5) over (2.7.0.2-5) ...
Processing triggers for mime-support (3.58) ...
# dpkg --configure man-db
Setting up man-db (2.7.0.2-5) ...
Updating database of manual pages ...
```

Обратите внимание: строки триггеров обработки (Processing triggers) относятся к коду, выполняемому автоматически, когда пакет добавляет, удаляет или изменяет файлы в некоторых контролируемых каталогах. Так, пакет *mime-support* контролирует каталог usr/lib/mime/packages и выполняет команду update-mime всякий раз, когда в этом каталоге что-то изменяется (например, /usr/lib/mime/packages/man-db в случае man-db).

Иногда у dpkg не получится установить пакет, и он выдает ошибку. Тем не менее вы можете приказать dpkg игнорировать ее и только вывести предупреждение с различными параметрами --force-*. Команда dpkg --force- help выведет на экран полный список этих параметров. Например, вы можете использовать dpkg для принудительной установки zsh:

```
$ dpkg -i --force-overwrite zsh_5.2-5+b1_amd64.deb
```

Популярная ошибка, с которой вы рано или поздно столкнетесь, — это конфликт файлов. Когда пакет содержит файл, уже установленный другим пакетом, dpkg откажется его установить. Появятся следующие типы сообщений:

```
Unpacking libgdm (from .../libgdm_3.8.3-2_amd64.deb) ...
dpkg: error processing /var/cache/apt/archives/libgdm_3.8.3-2_amd64.deb
      ➥(--unpack): trying to overwrite '/usr/bin/gdmflexiserver', which is
      ➥also in package gdm3 3.4.1-9
```

В таком случае, если считаете, что замена данного файла не представляет значительной угрозы для стабильности вашей системы (а как правило, это так), то можете использовать `--force-overwrite` для принудительной перезаписи файла.

Хотя есть множество доступных параметров `--force-*`, регулярно использовать можно только `--force-overwrite`. Эти параметры существуют для исключительных ситуаций, и лучше их не трогать максимально долго, чтобы соблюдать правила, установленные механизмом пакетирования. Не забывайте: эти правила обеспечивают согласованность и стабильность вашей системы.

Установка пакетов с помощью APT

Хотя APT является намного более продвинутым инструментом, чем dpkg, и делает гораздо больше работы, вы обнаружите, что взаимодействовать с пакетами с его помощью довольно просто. Вы можете добавить пакет в систему благодаря несложной команде `apt install` *пакет*. APT автоматически установит все необходимые зависимости:

```
# apt install kali-linux-gpu
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  oclgausscrack oclhashcat
The following NEW packages will be installed:
  kali-linux-gpu oclgausscrack oclhashcat
0 upgraded, 3 newly installed, 0 to remove and 416 not upgraded.
Need to get 2,494 kB of archives.
After this operation, 51.5 MB of additional disk space will be used.
Do you want to continue? [Y/n]
Get:1 http://archive-2.kali.org/kali kali-rolling/non-free amd64 oclhashcat
      ➥amd64 2.01+ git20160114-0kali2 [2,451 kB]
Get:2 http://archive-2.kali.org/kali kali-rolling/main amd64 oclgausscrack
      ➥amd64 1.3-1 kali2 [37.2 kB]
Get:3 http://archive-2.kali.org/kali kali-rolling/main amd64 kali-linux-gpu
      ➥amd64 2016.3.2 [6,412 B]
Fetched 2,494 kB in 0s (3,060 kB/s)
Selecting previously unselected package oclhashcat.
(Reading database ... 317084 files and directories currently installed.)
Preparing to unpack .../0-oclhashcat_2.01+git20160114-0kali2_amd64.deb ...
Unpacking oclhashcat (2.01+git20160114-0kali2) ...
Selecting previously unselected package oclgausscrack.
```

```
Preparing to unpack .../1-oclgausscrack_1.3-1kali2_amd64.deb ...
Unpacking oclgausscrack (1.3-1kali2) ...
Selecting previously unselected package kali-linux-gpu.
Preparing to unpack .../2-kali-linux-gpu_2016.3.2_amd64.deb ...
Unpacking kali-linux-gpu (2016.3.2) ...
Setting up oclhashcat (2.01+git20160114-0kali2) ...
Setting up oclgausscrack (1.3-1kali2) ...
Setting up kali-linux-gpu (2016.3.2) ...
```

Вы также можете использовать команды `apt-get install` *пакет* или `aptitude install` *пакет*. Для простой установки пакетов они делают практически то же самое. Как будет показано позже, отличия этих команд более заметны по отношению к обновлениям или когда разрешение зависимостей не имеет идеального решения.

Если файл `sources.list` перечисляет несколько дистрибутивов, то можете указать версию пакета с помощью команды `apt install` *пакет=версия*, но всегда желательно привести происхождение дистрибутива (kali-rolling, kali-dev или kali-bleeding- edge), воспользовавшись командой `apt install` *пакет/дистрибутив*.

Как и в случае с `dpkg`, вы можете приказать `apt` принудительно установить пакет и перезаписать файлы с помощью параметра `--force-overwrite`, но синтаксис в этом случае будет выглядеть немного странно, поскольку вы передаете аргумент через `dpkg`:

```
# apt -o Dpkg::Options::="--force-overwrite" install zsh
```

Обновление Kali Linux

Будучи дистрибутивом со скользящим релизом, Kali Linux обладает впечатляющими возможностями обновления. В этом подразделе мы рассмотрим, как легко обновить Kali, а также обсудим стратегии для планирования ваших обновлений.

Мы рекомендуем использовать регулярные обновления, поскольку с ними устанавливаются последние обновления безопасности. Чтобы начать процесс обновления, примените команду `apt update` и затем `apt upgrade`, `apt-get upgrade` или `aptitude safe-upgrade`. Эти команды отыскивают установленные пакеты, которые можно обновить без удаления каких-либо пакетов. Другими словами, цель состоит в том, чтобы обеспечить обновление с наименьшим вмешательством. Инструмент командной строки `apt-get` немного более требовательный, чем `aptitude` или `apt`, поскольку откажется устанавливать пакеты, не установленные ранее.

Инструмент `apt` обычно выбирает последнюю версию обновлений (за исключением пакетов с kali-bleeding-edge, которые по умолчанию игнорируются независимо от их версии).

Использовать конкретный дистрибутив при поиске обновленных пакетов поможет обращение к параметру `-t` или `--target-release`, за которым следует имя нужного дистрибутива (например, `apt -t kali-rolling upgrade`). Чтобы не указывать это каждый раз, когда вы используете `apt`, можете добавить строку `APT :: Default-Release "kali-rolling";` в файл `/etc/apt/apt.conf.d/local`.

Для более важных обновлений, таких как обновление основной версии, используйте команду `apt full-upgrade`. При ее выполнении `apt` завершит обновление, даже если ему нужно удалить некоторые устаревшие пакеты или установить новые зависимости. Кроме того, данную команду нужно применять для регулярных обновлений вашей системы Kali Rolling. Это действие настолько простое, что вряд ли требует каких-либо дополнительных разъяснений: популярность APT основывается именно на этой замечательной функциональности.

В отличие от `apt` и `aptitude`, `apt-get` не распознает команду `full-upgrade`. Вместо этого вы должны использовать `apt-get dist-upgrade` (обновление дистрибутива), известную команду, которую `apt` и `aptitude` также принимают для обратной совместимости.

Будьте всегда в курсе важных изменений

Чтобы предвидеть некоторые из этих проблем, вы можете установить пакет apt-listchanges, отображающий информацию о возможных проблемах в начале обновления пакета. Данная информация собирается составителями пакетов и помещается в файлы /usr/share/doc/package/NEWS.Debian. Их внимательное чтение (к примеру, с помощью apt-listchanges) должно помочь вам избежать неприятных сюрпризов.

Ввиду того что Kali является дистрибутивом со скользящим релизом, он получает обновления несколько раз в день. Однако это не всегда лучшая стратегия. Итак, насколько часто вам необходимо обновлять Kali Linux? Безусловно, не существует каких-либо строгих правил, но есть ряд основных принципов, которые могут вам помочь определиться. Выполнять обновления нужно в следующих случаях:

- когда вам известно о проблеме безопасности, исправленной в обновлении;
- если вы подозреваете, что обновленная версия может исправить ошибку, с которой вы столкнулись;
- прежде чем сообщить об ошибке, чтобы убедиться в ее присутствии в доступной вам последней версии;
- нередко для получения обновлений безопасности, о которых вы не слышали.

Кроме того, существуют ситуации, в которых лучше не выполнять обновление. Так, ниже представлены случаи, когда мы не рекомендуем обновлять систему.

- Если у вас не будет достаточно времени, чтобы исправить возможные неполадки (например, вы будете офлайн или собираетесь подготовить презентацию на своем компьютере); лучше выполнить обновление позже, когда у вас будет достаточно времени для устранения проблемы, возникшей в процессе.
- Если в последнее время у вас произошло (или продолжается) нарушение работы и вы опасаетесь, что еще не все проблемы разрешены. Например, при выпуске новой версии GNOME не все пакеты обновляются одновременно, и у вас, вероятно, будет сочетание пакетов старой и новой версий. В большинстве случаев

это нормально и позволяет выпускать обновления постепенно, но всегда существуют исключения, и работа некоторых приложений может быть нарушена из-за таких несоответствий.

- Если вывод команды `apt full-upgrade` сообщает, что пакеты, необходимые для вашей работы, будут удалены. В подобных случаях вам следует разобраться в ситуации и попытаться понять, почему `apt` хочет их удалить. Возможно, пакеты в настоящее время повреждены, и, следовательно, придется подождать, пока будут доступны исправленные версии. Или же пакеты просто-напросто устарели, и вам нужно определить, чем их заменить, а затем продолжить полное обновление.

В общем, мы рекомендуем обновлять Kali не реже одного раза в неделю. Вы можете, конечно, обновляться ежедневно, но мы считаем, что это не имеет смысла. Даже если зеркала синхронизируются четырежды в день, обновления от Debian обычно поступают только один раз в день.

Удаление и очистка пакетов

Удалить пакет еще проще, чем установить его. Рассмотрим, как удалить пакет с помощью инструментов `dpkg` и `apt`.

Чтобы удалить пакет с помощью `dpkg`, укажите параметр `-r` или `--remove`, а затем имя пакета. Однако на этом удаление не завершено: все файлы конфигурации, сценарии поддержки, файлы журналов (системные журналы), информация, созданная демоном (скажем, содержимое каталога сервера LDAP или базы данных для SQL-сервера), и большинство других данных пользователя, обрабатываемых пакетом, остаются нетронутыми. Параметр `remove` позволяет легко удалить программу, а затем переустановить ее с той же конфигурацией. Помните, что зависимости также не удаляются. Рассмотрим пример:

```
# dpkg --remove kali-linux-gpu
(Reading database ... 317681 files and directories currently installed.)
Removing kali-linux-gpu (2016.3.2) ...
```

Вы также можете удалить пакеты из системы с помощью команды `apt remove` *`пакет`*. APT автоматически удалит пакеты, которые зависят от удаленного пакета. Как и в примере `dpkg`, файлы конфигурации и данные пользователя не будут удалены.

Добавляя суффиксы к именам пакетов, вы можете применять `apt` (или `apt-get` или `aptitude`) для установки определенных пакетов и удаления других в одной и той же командной строке. Используя команду `apt install`, добавьте `-` к именам пакетов, которые хотите удалить. С помощью команды `apt remove` добавьте `+` к именам пакетов, которые хотите установить.

Следующий пример показывает два различных способа установки *`пакета1`* и удаления *`пакета2`*.

```
# apt install пакет1 пакет2-
[...]
# apt remove пакет1+ пакет2
[...]
```

Этот способ также можно использовать, чтобы исключить пакеты, которые вы не хотите устанавливать, например ввиду рекомендаций (мы обсудим данный вопрос немного позже). В общем, программа решения зависимостей задействует эту информацию в качестве подсказки для поиска альтернативных решений.

Для удаления всех данных, связанных с каким-либо пакетом, вы можете применить команды `dpkg -P` *пакет* или `apt purge` *пакет*. Они полностью удалят пакет и все данные пользователя и в случае с `apt` также удалят и все зависимости.

```
# dpkg -r debian-cd
(Reading database ... 97747 files and directories currently installed.)
Removing debian-cd (3.1.17) ...
# dpkg -P debian-cd
(Reading database ... 97401 files and directories currently installed.)
Removing debian-cd (3.1.17) ...
Purging configuration files for debian-cd (3.1.17) ...
```

Внимание! Учитывая окончательный характер очистки, дважды подумайте, прежде чем ее применить. Вы потеряете все, что связано с этим пакетом.

Проверка пакетов

Рассмотрим ряд инструментов для проверки пакетов Debian. В частности, остановимся на командах `dpkg`, `apt` и `apt-cache`, которые используются для запроса и визуализации базы данных пакета.

Запрос базы данных dpkg и проверка файлов с расширением .deb

Начнем с обзора нескольких параметров `dpkg`, которые запрашивают внутреннюю базу данных `dpkg`. Она находится в файловой системе в каталоге `/var/lib/dpkg` и содержит несколько разделов, включающих сценарии конфигурации (`/var/lib/dpkg/info`), список файлов, установленных пакетом (`/var/lib/dpkg/info/*.list`), и статус каждого установленного пакета (`/var/lib/dpkg/status`). Вы можете использовать `dpkg` для взаимодействия с файлами в этой базе данных. Обратите внимание: большинство параметров доступны как в их длинной версии (одно или несколько релевантных слов с двойным тире вначале), так и в короткой (одно тире и одна буква, чаще всего первая буква слова из длинной версии). Это условное обозначение настолько популярно, что стало стандартом POSIX.

Для начала рассмотрим параметр `--listfiles` *пакет* (или `-L`), выводящий на экран список файлов, которые были установлены указанным пакетом:

```
$ dpkg -L base-passwd
/.
/usr
/usr/sbin
/usr/sbin/update-passwd
/usr/share
```

```
/usr/share/lintian
/usr/share/lintian/overrides
/usr/share/lintian/overrides/base-passwd
/usr/share/doc-base
/usr/share/doc-base/users-and-groups
/usr/share/base-passwd
/usr/share/base-passwd/group.master
/usr/share/base-passwd/passwd.master
/usr/share/man
/usr/share/man/pl
/usr/share/man/pl/man8
/usr/share/man/pl/man8/update-passwd.8.gz
[...]
/usr/share/doc
/usr/share/doc/base-passwd
/usr/share/doc/base-passwd/users-and-groups.txt.gz
/usr/share/doc/base-passwd/changelog.gz
/usr/share/doc/base-passwd/copyright
/usr/share/doc/base-passwd/README
/usr/share/doc/base-passwd/users-and-groups.html
```

Команда `dpkg --search` *файл* (или `-S`) находит все пакеты, которые включает файл или путь, передаваемый в аргументе. Например, для поиска пакета, содержащего путь `/bin/date`, нужно выполнить команду:

```
$ dpkg -S /bin/date
coreutils: /bin/date
```

Команда `dpkg --status` *пакет* (или `-s`) отображает заголовок установленного пакета. Так, для поиска заголовков пакета `coreutils` нужно выполнить следующую команду:

```
$ dpkg -s coreutils
Package: coreutils
Essential: yes
Status: install ok installed
Priority: required
Section: utils
Installed-Size: 13855
Maintainer: Michael Stone <mstone@debian.org>
Architecture: amd64
Multi-Arch: foreign
Version: 8.23-3
Replaces: mktemp, realpath, timeout
Pre-Depends: libacl1 (>= 2.2.51-8), libattr1 (>= 1:2.4.46-8), libc6 (>= 2.17),
    ➥ libselinux1 (>= 2.1.13)
Conflicts: timeout
Description: GNU core utilities
This package contains the basic file, shell and text manipulation
utilities which are expected to exist on every operating system.
.
Specifically, this package includes:
```

```
arch base64 basename cat chcon chgrp chmod chown chroot cksum comm cp
csplit cut date dd df dir dircolors dirname du echo env expand expr
factor false flock fmt fold groups head hostid id install join link ln
logname ls md5sum mkdir mkfifo mknod mktemp mv nice nl nohup nproc numfmt
od paste pathchk pinky pr printenv printf ptx pwd readlink realpath rm
rmdir runcon sha*sum seq shred sleep sort split stat stty sum sync tac
tail tee test timeout touch tr true truncate tsort tty uname unexpand
uniq unlink users vdir wc who whoami yes
Homepage: http://gnu.org/software/coreutils
```

Команда `dpkg --list` (или `-l`) отображает список известных системе пакетов и их статус установки. Вы также можете использовать параметр `grep` для поиска определенных полей или задать шаблон (например, `b*`) для поиска пакетов, которые соответствуют определенной строке частичного поиска. Так, чтобы вывести на экран список всех пакетов, начинающихся с `'b'`, примените команду:

```
$ dpkg -l 'b*'
Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Inst/Conf-files/Unpacked/halF-conf/Half-inst/trig-aWait/Trig-pend
|/ Err?=(none)/Reinst-required (Status,Err: uppercase=bad)
||/ Name           Version      Architecture Description
+++-==============-============-============-=================================
ii  b43-fwcutter   1:019-3      amd64        utility for extracting Broadcom 4
ii  backdoor-facto 3.4.2-0kali1 all          Patch win32/64 binaries with shel
un  backupninja    <none>       <none>       (no description available)
un  backuppc       <none>       <none>       (no description available)
ii  baobab         3.22.1-1     amd64        GNOME disk usage analyzer
[...]
```

Команда `dpkg --contents` *файл*`.deb` (или `-c`) перечисляет все файлы, содержащиеся в заданном файле формата `.deb`:

```
$ dpkg -c /var/cache/apt/archives/gnupg_1.4.18-6_amd64.deb
drwxr-xr-x root/root         0 2014-12-04 23:03 ./
drwxr-xr-x root/root         0 2014-12-04 23:03 ./lib/
drwxr-xr-x root/root         0 2014-12-04 23:03 ./lib/udev/
drwxr-xr-x root/root         0 2014-12-04 23:03 ./lib/udev/rules.d/
-rw-r--r-- root/root      2711 2014-12-04 23:03 ./lib/udev/rules.d/60-gnupg.rules
drwxr-xr-x root/root         0 2014-12-04 23:03 ./usr/
drwxr-xr-x root/root         0 2014-12-04 23:03 ./usr/lib/
drwxr-xr-x root/root         0 2014-12-04 23:03 ./usr/lib/gnupg/
-rwxr-xr-x root/root     39328 2014-12-04 23:03 ./usr/lib/gnupg/gpgkeys_ldap
-rwxr-xr-x root/root     92872 2014-12-04 23:03 ./usr/lib/gnupg/gpgkeys_hkp
-rwxr-xr-x root/root     47576 2014-12-04 23:03 ./usr/lib/gnupg/gpgkeys_finger
-rwxr-xr-x root/root     84648 2014-12-04 23:03 ./usr/lib/gnupg/gpgkeys_curl
-rwxr-xr-x root/root      3499 2014-12-04 23:03 ./usr/lib/gnupg/gpgkeys_mailto
drwxr-xr-x root/root         0 2014-12-04 23:03 ./usr/bin/
-rwxr-xr-x root/root     60128 2014-12-04 23:03 ./usr/bin/gpgsplit
-rwxr-xr-x root/root   1012688 2014-12-04 23:03 ./usr/bin/gpg
[...]
```

Команда `dpkg --info` *файл*`.deb` (или `-I`) выведет на экран заголовки указанного файла формата `.deb`:

```
$ dpkg -I /var/cache/apt/archives/gnupg_1.4.18-6_amd64.deb
 new debian package, version 2.0.
 size 1148362 bytes: control archive=3422 bytes.
    1264 bytes,    26 lines      control
    4521 bytes,    65 lines      md5sums
     479 bytes,    13 lines   *  postinst             #!/bin/sh
     473 bytes,    13 lines   *  preinst              #!/bin/sh
 Package: gnupg
 Version: 1.4.18-6
 Architecture: amd64
 Maintainer: Debian GnuPG-Maintainers <pkg-gnupg-maint@lists.alioth.debian.org>
 Installed-Size: 4888
 Depends: gpgv, libbz2-1.0, libc6 (>= 2.15), libreadline6 (>= 6.0),
    ➥ libusb-0.1-4 (>= 2:0.1.12), zlib1g (>= 1:1.1.4)
 Recommends: gnupg-curl, libldap-2.4-2 (>= 2.4.7)
 Suggests: gnupg-doc, libpcsclite1, parcimonie, xloadimage | imagemagick | eog
 Section: utils
 Priority: important
 Multi-Arch: foreign
 Homepage: http://www.gnupg.org
 Description: GNU privacy guard - a free PGP replacement
  GnuPG is GNU's tool for secure communication and data storage.
  It can be used to encrypt data and to create digital signatures.
  It includes an advanced key management facility and is compliant
  with the proposed OpenPGP Internet standard as described in RFC 4880.
[...]
```

Кроме того, `dpkg` можно использовать для сравнения версий пакетов с помощью параметра `--compare-versions`, часто вызываемого внешними программами, включая сценарии конфигураций, которые выполняются самим `dpkg`. Для этого параметра требуются три аргумента: номер версии, оператор сравнения и номер другой версии. Существуют различные возможные операторы: `lt` (строго меньше), `le` (меньше или равно), `eq` (равно), `ne` (не равно), `ge` (больше или равно) и `gt` (строго больше). При правильном сравнении `dpkg` возвращает `0` (успех); в противном случае возвращаемое значение будет не равно нулю (что означает неверное сравнение). Рассмотрим эти сравнения:

```
$ dpkg --compare-versions 1.2-3 gt 1.1-4
$ echo $?
0
$ dpkg --compare-versions 1.2-3 lt 1.1-4
$ echo $?
1
$ dpkg --compare-versions 2.6.0pre3-1 lt 2.6.0-1
$ echo $?
1
```

Обратите внимание на неожиданный сбой последнего сравнения: для `dpkg` строка `pre` (обычно обозначающая предварительную версию) не имеет конкретного значения, и `dpkg` просто интерпретирует ее как строку, и в этом случае при сравнении по алфавитному порядку значение `2.6.0pre3-1` больше, чем `2.6.0-1`. Если нужно, чтобы номер версии пакета указывал на предварительность релиза, то используют символ тильды ~:

```
$ dpkg --compare-versions 2.6.0~pre3-1 lt 2.6.0-1
$ echo $?
0
```

Запрос к базе данных на наличие доступных пакетов с помощью команд apt-cache и apt

Команда `apt-cache` может отображать большую часть информации, хранящейся во внутренней базе данных APT. Данная информация является своего рода кэшем, поскольку собирается из разных источников, перечисленных в файле `sources.list`. Это происходит во время выполнения операции `apt update`.

> Словарь терминов
>
> **Кэш**
>
> Кэш — система временного хранения, служащая для ускорения доступа к часто используемым данным, когда обычный метод доступа является затратным (по отношению к производительности). Эта идея применима во многих ситуациях и в разных масштабах: от ядра микропроцессоров до объемных систем хранения информации.
>
> В случае с APT ссылочные файлы Packages расположены на зеркалах Debian. Учитывая сей факт, было бы очень неэффективно проводить каждый поиск через онлайн-базы данных пакетов. Вот почему APT хранит копию этих файлов (в каталоге /var/lib/apt/lists/) и поиск выполняется в этих локальных файлах. Аналогично /var/cache/apt/archives/ содержит кэшированную копию уже скачанных пакетов, чтобы обойтись без повторного скачивания при необходимости переустановить их.
>
> Чтобы избежать чрезмерного использования диска при частом обновлении, вы должны регулярно сортировать каталог /var/cache/apt/archives/. Для этого можно применить две команды: apt clean (или apt-get clean), которая полностью очищает каталог; apt autoclean (или apt-get autoclean), удаляющая только те пакеты, которые больше нельзя скачать, поскольку они исчезли с зеркала и поэтому являются бесполезными. Обратите внимание: для предотвращения удаления файлов формата .deb, установленных в настоящее время, служит параметр конфигурации APT::Clean-Installed. И еще: apt удаляет скачанные файлы после их установки. Это важно помнить во время использования других инструментов.

Команда `apt-cache` может выполнять поиск пакетов по ключевым словам с помощью `apt-cache search` *`ключевое слово`*. Она также способна отображать

заголовки доступных версий пакета путем `apt-cache show` *`пакет`*. Эта команда выводит описание пакета, его зависимости и имя разработчика. Данная функция довольно полезна при определении пакетов, которые устанавливаются через метапакеты, такие как `kali-linux-wireless`, `kali-linux-web` и `kali-linux-gpu`. Обратите внимание: команды `apt search`, `apt show`, `aptitude search` и `aptitude show` работают одинаково.

Альтернатива: axi-cache

Инструмент apt-cache search довольно примитивен. В основном он использует параметр grep для поиска в описаниях пакетов. Очень часто возвращает либо слишком много результатов, либо вообще ничего, если применяется большое количество ключевых слов.

С другой стороны, команда axi-cache search показывает лучшие результаты, отсортированные по степени важности. Она задействует поисковую систему Xapian и является частью пакета apt-xapian-index, который индексирует всю информацию о пакете (и многое другое, например файлы формата .desktop из всех пакетов Debian). Команда работает с тегами и предоставляет результаты за считаные миллисекунды.

```
$ axi-cache search forensics graphical
5 results found.
Results 1-5:
100 % autopsy - graphical interface to SleuthKit
82 % forensics-colorize - show differences between files using
      ➥color graphics
73 % dff - Powerful, efficient and modular digital forensic
      ➥framework
53 % gpart - Guess PC disk partition table, find lost
      ➥partitions
46 % testdisk - Partition scanner and disk recovery tool, and
      ➥PhotoRec file recovery tool
More terms: colorize partitions file disklabel autopsy digital
      ➥differences
More tags: admin::forensics security::forensics role::program
      ➥admin::recovery interface::commandline admin::boot
      ➥scope::utility
```

Иные функции используются реже. Скажем, команда `apt-cache policy` отображает приоритеты как источников пакетов, так и отдельных пакетов. Другим примером является команда `apt-cache dumpa vail`, отображающая заголовки всех доступных версий всех пакетов. Команда `apt-cache pkgnames` отображает список всех пакетов, которые встречаются в кэше хотя бы раз.

Устранение проблем

Рано или поздно у вас могут возникнуть определенные проблемы во время взаимодействия с пакетами. В данном подразделе мы постараемся описать основные шаги, которые необходимо будет предпринять для устранения проблем, а также

расскажем о ряде инструментов, способных приблизить к разрешению той или иной проблемы.

Проблемы с обработкой после обновления

Несмотря на усердную работу создателей Kali/Debian над тем, чтобы обновление системы проходило безболезненно, не всегда оно выполняется так гладко, как ожидается. Новые версии программного обеспечения могут быть несовместимы с предыдущими (например, их поведение по умолчанию или их формат данных можно изменить), или же некоторые ошибки предыдущих версий могут сохраняться в новой версии, невзирая на тестирование, проводимое разработчиками пакетов и пользователями Debian Unstable.

Применение отчетов об ошибке. Иногда бывает, что новая версия программного обеспечения вообще не работает. Такое обычно случается, если приложение не особо популярно и было недостаточно протестировано. Первое, что нужно сделать, — обратиться к системам отслеживания ошибок Kali и Debian (https://www.debian.org/Bugs/) по адресу https://bugs.debian.org/пакет и проверить, сообщал ли кто-нибудь ранее об этой проблеме. При отсутствии отчетов о данной ошибке вам следует составить его самостоятельно (см. раздел 6.3). Если отчет уже был подан до вас, то он и все связанные с ним сообщения — отличный источник информации относительно самой ошибки. В одних случаях патч, исправляющий ошибку, уже существует и является доступным в самом отчете об ошибке; вы можете перекомпилировать исправленную версию неисправного пакета локально (см. раздел 9.1). В других случаях пользователи могли найти некий искусный метод или обходной путь для работы с данной проблемой и поделились им в своих комментариях к отчету; подобного рода инструкции помогут вам разобраться с возникшей проблемой, пока не выйдет соответствующий патч. В идеале ошибку пакета могли уже исправить, и вы можете найти информацию об этом в отчете об ошибке.

Откат на предыдущую рабочую версию. Когда проблемой является очевидная регрессия (при рабочей прежней версии), вы можете попытаться использовать предыдущую версию пакета. В этом случае понадобится копия старой версии. Если у вас есть доступ к такой версии в одном из репозиториев, настроенных в APT, то можете применить простую однострочную команду для отката версии (см. пункт «Установка пакетов с помощью APT» подраздела «Установка пакетов» раздела 8.2). Но с плавающим релизом Kali вы обычно найдете только одну версию для каждого пакета.

Вы все равно можете попытаться найти старый файл формата `.deb` и установить его вручную с помощью `dpkg`. Старые файлы формата `.deb` можно найти в нескольких местах:

- в кэше APT в каталоге `/var/cache/apt/archives/`;
- в каталоге `pool` на вашем стандартном зеркале Kali (удаленные и устаревшие пакеты хранятся в течение трех-четырех дней, чтобы избежать проблем с пользователями, не имеющими последних индексов пакета);

- по адресу http://snapshot.debian.org/, если поврежденный пакет был предоставлен Debian, а не Kali; этот сервис хранит абсолютно все версии пакетов Debian.

Работа с поврежденными сценариями поддержки. Иногда обновление прерывается, поскольку один из сценариев поддержки пакета не работает (обычно это `postinst`). В подобных случаях вы можете попытаться диагностировать проблему и, вероятно, обойти ее, отредактировав проблемный сценарий.

Здесь мы полагаемся на факт, что сценарий поддержки хранится в каталоге `/var/lib/dpkg/info/` и мы можем просмотреть и изменить его.

Поскольку сценарии поддержки обычно являются простыми сценариями оболочки, то мы можем вставить строку `set -x` сразу после шебанга и перезапустить их (с помощью команды `dpkg --configure -a` для сценария `postinst`) с целью увидеть, что именно происходит и в чем заключается ошибка. Вывод этой команды также способен прекрасно дополнить любой ваш отчет об ошибке.

С этими знаниями вы можете как исправлять исходную проблему, так и преобразовать неисправную команду в рабочую (скажем, путем добавления `|| true` в конце строки).

Обратите внимание: данная методика не работает в случае сбоя сценария `preinst`, поскольку он выполняется еще до установки пакета, поэтому его еще нет в его конечном месте размещения. Методика действует для сценариев `postrm` и `prerm`, хотя вам нужно будет удалить пакет (и, соответственно, обновить), чтобы запустить их.

Файл регистрации инструмента dpkg

Инструмент `dpkg` хранит записи всех своих действий в журнале `/var/log/dpkg.log`. Последний является чрезвычайно подробным, так как детально описывает все статусы каждого пакета. Помимо того что он позволяет отслеживать поведение `dpkg`, он помогает сохранить историю разработки системы: вы можете узнать точный момент установки или обновления любого из пакетов, и эта информация может быть чрезвычайно полезна для понимания причины внезапного изменения поведения. Кроме того, при записи всех версий можно легко перекрестно проверить информацию с помощью файла `changelog.Debian.gz` для пакетов, которые вызывают вопросы, или даже применяя онлайн-отчеты об ошибках.

```
# tail /var/log/dpkg.log
2016-12-22 09:04:05 status installed kali-linux-gpu:amd64 2016.3.2
2016-12-22 09:20:07 startup packages remove
2016-12-22 09:20:07 status installed kali-linux-gpu:amd64 2016.3.2
2016-12-22 09:20:07 remove kali-linux-gpu:amd64 2016.3.2 <none>
2016-12-22 09:20:07 status half-configured kali-linux-gpu:amd64 2016.3.2
2016-12-22 09:20:07 status half-installed kali-linux-gpu:amd64 2016.3.2
2016-12-22 09:20:07 status config-files kali-linux-gpu:amd64 2016.3.2
2016-12-22 09:20:07 status config-files kali-linux-gpu:amd64 2016.3.2
2016-12-22 09:20:07 status config-files kali-linux-gpu:amd64 2016.3.2
2016-12-22 09:20:07 status not-installed kali-linux-gpu:amd64 <none>
```

Переустановка пакетов с помощью команд apt --reinstall и aptitude reinstall

Когда вы случайно наносите ущерб своей системе, удаляя или изменяя определенные файлы, самый простой способ их восстановить — переустановить поврежденный пакет. К сожалению, система пакетирования обнаруживает, что пакет уже установлен, и вежливо отказывается переустанавливать его. Во избежание этого используйте параметр `--reinstall` для команд `apt` и `apt-get`. Следующая команда переустановит пакет `postfix`, даже если он уже существует в системе:

```
# apt --reinstall install postfix
```

Команда `aptitude` немного отличается, но тем не менее приводит к тому же результату, если применять ее как `aptitude reinstall postfix`. Команда `dpkg` не предотвращает переустановку, но редко используется напрямую.

> **Не используйте команду apt --reinstall для восстановления после атаки**
>
> Применение команды apt --reinstall для восстановления пакетов, измененных во время атаки, не приведет к восстановлению системы в ее первоначальном виде.
>
> После атаки вы не должны полагаться абсолютно ни на что: возможно, dpkg и apt были заменены вредоносными программами и не будут переустанавливать файлы так, как вы того ожидаете. Кроме того, злоумышленник мог изменить или создать файлы вне контроля dpkg.

Помните: вы можете указать конкретный дистрибутив с помощью `apt`; это позволяет вам откатиться к более старой версии пакета (если, к примеру, вы уверены в ее работе должным образом) при условии, что она по-прежнему доступна в одном из источников, указанных в файле `sources.list`:

```
# apt install w3af/kali-rolling
```

Использование параметра --force-* для восстановления поврежденных зависимостей

Неосторожное использование параметра `--force-*` или какие-либо другие неисправности могут привести к тому, что команды APT перестанут выполнять свои функции в системе. По сути, некоторые из этих параметров разрешают установку пакета, даже если зависимость не выполняется или существует конфликт в системе. Результатом является непоследовательная система с точки зрения зависимостей, и команды APT откажутся совершать любые действия, кроме тех, которые вернут систему в исправное состояние (зачастую это установка отсутствующей зависимости или удаление проблемного пакета). О неисправности обычно сигнализирует сообщение, подобное представленному ниже. Оно получено после установки новой

версии `rdesktop`, во время которой игнорировалась ее зависимость от более новой версии `libc6`:

```
# apt full-upgrade
[...]
You might want to run 'apt-get -f install' to correct these.
The following packages have unmet dependencies:
rdesktop: Depends: libc6 (>= 2.5) but 2.3.6.ds1-13etch7 is installed
E: Unmet dependencies. Try using -f.
```

Если же вы — отважный администратор, уверенный в правильности своего анализа, то можете игнорировать зависимость или конфликт и использовать параметр `--force-*`. В этом случае при желании продолжать задействовать `apt` или `aptitude` вам нужно отредактировать файл `/var/lib/dpkg/status`, чтобы удалить или изменить зависимость или конфликт, которые вы решили переопределить.

Этот прием считается дурным тоном и никогда не должен использоваться, за исключением самых крайних случаев. Обычно более подходящим решением является перекомпиляция проблемного пакета или использование его новой версии (потенциально исправленной) из репозитория, предоставляющего адаптации (новые версии, специально перекомпилированные для работы в более старой среде).

Пользовательские интерфейсы: aptitude и synaptic

APT — программа, работающая на C++, код которой в основном находится в общей библиотеке `libapt-pkg`. Именно эта общая библиотека открывает возможности для создания пользовательских интерфейсов (фронтэндов), поскольку код, доступный в общей библиотеке, другие разработчики могут использовать повторно. Изначально `apt-get` был разработан лишь как тестовый интерфейс для `libapt-pkg`, но ввиду его ошеломляющего успеха данный факт умалчивается.

Со временем, несмотря на популярность интерфейсов командной строки, таких как `apt` и `apt-get`, были разработаны различные графические интерфейсы. В этом подразделе мы рассмотрим два из них: `aptitude` и `synaptic`.

Aptitude

Программа aptitude (рис. 8.1), представляет собой интерактивную программу, которая может применяться в полуграфическом режиме на консоли. Вы можете просмотреть список установленных и доступных пакетов, получить всю информацию о них и выбрать пакеты для установки или удаления. Программа разработана специально для использования администраторами, поэтому ее поведение по умолчанию намного более интеллектуально, чем APT, а интерфейс значительно понятнее.

После запуска `aptitude` на экран будет выведен список пакетов, отсортированных по состоянию (установленные, неустановленные или установленные, но недоступные на зеркалах), тогда как в других разделах отображаются задачи, виртуальные

пакеты и новые пакеты, появившиеся недавно на зеркалах. Для облегчения тематического просмотра доступны разнообразные режимы.

```
 Actions  Undo  Package  Resolver  Search  Options  Views  Help
C T: Menu  ?: Help  q: Quit  u: Update  g: Download/Install/Remove Pkgs
aptitude 0.6.11              Will use 6,202 kB of disk spac DL Size: 2,765 kB
--\ Installed Packages (270)
  --\ admin - Administrative utilities (install software, manage users, etc) (43
    --\ main - The main Debian archive (43)
i A   acpi support base                              0.142 6        0.142 6
i     acpid                                          1:2.0.23-2     1:2.0.23-2
i A   adduser                                        3.113+nmu3     3.113+nmu3
i A   apt                                            1.0.9.6        1.0.9.6
i A   apt-utils                                      1.0.9.6        1.0.9.6
i     aptitude                                       0.6.11 1+b1    0.6.11 1+b1
i A   aptitude-common                                0.6.11-1       0.6.11-1
terminal-based package manager
aptitude is a package manager with a number of useful features, including: a  #
mutt-like syntax for matching packages in a flexible manner, dselect-like
persistence of user actions, the ability to retrieve and display the Debian
changelog of most packages, and a command-line mode similar to that of apt-get.

aptitude is also Y2K-compliant, non-fattening, naturally cleansing, and
housebroken.
Homepage: http://aptitude.alioth.debian.org/


Tags: admin::configuring, admin::package-management, implemented-in::c++,
```

Рис. 8.1. Менеджер пакетов aptitude

Во всех случаях `aptitude` отображает список, объединяющий на экране категории и пакеты. Категории организованы в древовидную структуру, ветви которой можно развернуть или свернуть с помощью клавиши Enter. Клавишу + следует использовать для обозначения пакетов для установки, клавишей - отмечают пакеты к удалению, а _ применяют для их очистки. Обратите внимание: эти клавиши можно задействовать и по отношению к категориям; в таком случае соответствующие действия будут применяться ко всем пакетам категории. Клавиша u обновляет списки доступных пакетов, в то время как сочетание клавиш Shift+u подготавливает полное системное обновление. Клавиша g переключается на сводный режим просмотра запрошенных изменений (повторный ввод g применит изменения), а клавиша q завершает текущее представление. Если вы находитесь в исходном режиме просмотра, то это закроет `aptitude`.

> **Документация aptitude**
>
> В этом пункте не рассматриваются более тонкие нюансы использования aptitude, он скорее фокусируется на предоставлении пользователю самой необходимой информации. Программа aptitude достаточно хорошо документирована, и мы рекомендуем познакомиться с полным руководством, которое доступно в пакете aptitude-doc-en (file:///usr/share/doc/aptitude/html/en/index.html).

Для поиска пакета вы можете ввести / и далее шаблон поиска. Этот шаблон соответствует имени пакета, но может применяться и к описанию (если ему предшествует ~d), к разделу (~s) или другим характеристикам, указанным в документации.

Те же шаблоны способны фильтровать список отображаемых пакетов: нажмите клавишу `l` (от limit — «ограничение») и введите шаблон.

Управление *автоматической меткой* пакетов Debian (см. подраздел «Отслеживание автоматически установленных пакетов» раздела 8.3) выполняется очень просто благодаря `aptitude`. Вы можете просматривать список установленных пакетов и отмечать пакеты как автоматические с помощью сочетания клавиш Shift+m или удалять отметку, задействуя клавишу `m`. Автоматические пакеты отображаются в списке пакетов с пометкой `A`. Эта функция также предлагает простой способ визуализации пакетов, применяемых машиной, без их библиотек и зависимостей, которые вам абсолютно не интересны. Дополнительный шаблон, пригодный для использования с клавишей `l` (для активации режима фильтра), — `~i!~M`. Он обозначает, что вы хотите видеть только установленные пакеты (`~i`), не отмеченные как автоматические (`!~M`).

Использование aptitude в интерфейсе командной строки

Большинство функций Aptitude доступно как через интерактивный интерфейс, так и через командную строку.

Эти команды покажутся очень знакомыми постоянным пользователям apt-get и apt-cache.

Расширенные функции aptitude также доступны в командной строке. Вы можете использовать те же шаблоны поиска пакетов, что и в интерактивной версии. Например, если хотите очистить список установленных вручную пакетов и при этом знаете, что ни одна из локально установленных программ не требует каких-либо конкретных библиотек или модулей Perl, то можете пометить соответствующие пакеты как автоматические с помощью следующей команды:

```
# aptitude markauto '~slibs|~sperl'
```

Этот пример прекрасно демонстрирует всю силу системы шаблонов поиска aptitude, которая позволяет мгновенно выбрать все пакеты в разделах libs и perl.

Остерегайтесь случаев, когда пакеты отмечены как автоматические и при этом никакие другие пакеты не зависят от них. Такие пакеты будут немедленно удалены (после запроса на подтверждение).

Управление рекомендациями, предложениями и задачами. Еще одна интересная особенность программы `aptitude` заключается в том, что последняя учитывает рекомендации между пакетами, но при этом предоставляет пользователям возможность не устанавливать их в определенных случаях. Например, пакет *gnome* рекомендует (среди прочих) пакет *gdebi*. Когда вы выбираете первый для установки, рекомендуемый также будет выбран (и отмечен как автоматический, если он еще не установлен в системе). Это можно увидеть, нажав клавишу `g`: пакет *gdebi* появится в сводке отложенных действий в списке пакетов, установленных автоматически для удовлетворения зависимостей. Однако вы можете не устанавливать его, отменив выбор пакета перед подтверждением операций.

Обратите внимание: эта функция отслеживания рекомендаций не применяется к обновлениям. Так, если новая версия *gnome* рекомендует пакет, который ранее не рекомендовался, то пакет не подвергнется отметке для установки. Однако он будет указан на экране обновления, чтобы администратор в случае необходимости мог выбрать его для установки.

Кроме того, учитываются предложения между пакетами, но в соответствии с их конкретным статусом. Например, когда *gnome* предлагает *dia-gnome*, последний будет отображаться в сводке отложенных действий (в разделе пакетов, предложенных другими пакетами). Таким образом, администратор сможет решить, принимать ли во внимание данное предложение. Поскольку это всего лишь предложение, а не зависимость или рекомендация, пакет не будет выбран автоматически — его выбор осуществляется вручную (то есть пакет не получит отметку «автоматический»).

И еще помните: `aptitude` разумно использует концепцию задач. Поскольку задачи отображаются в виде категорий на экранах списков пакетов, то вы можете как выбрать всю задачу для установки или удаления, так и просмотреть список пакетов, включенных в нее, чтобы использовать некоторые из них.

Более эффективные алгоритмы. В заключение отметим, что `aptitude` имеет более сложные алгоритмы по сравнению с `apt`, когда дело доходит до решения сложных ситуаций. Когда запрашивается несколько действий, совместное выполнение которых приведет к неполадкам в системе, `aptitude` оценивает несколько вероятных сценариев и представляет их в порядке уменьшения важности. Однако эти алгоритмы не являются надежными. К счастью, всегда есть возможность вручную выбрать действия для выполнения. Когда текущие выбранные действия приводят к противоречиям, в верхней части экрана указывается количество неисправных пакетов (вы можете перейти к этим пакетам, нажав клавишу b). Затем вы можете вручную создать решение, в частности получить доступ к различным доступным версиям, указав пакет с помощью клавиши Enter. Если выбор одной из этих версий решает проблему, то не стоит сомневаться в верности решения. Когда количество неисправных пакетов снижается до нуля, можете без опасений перейти к сводке отложенных действий для последней проверки перед их подтверждением.

Журнал Aptitude

Как и dpkg, aptitude хранит записи о выполненных действиях в своем журнале (/var/log/aptitude). Но поскольку команды работают на совершенно разных уровнях, то информация в их журналах будет значительно отличаться. В то время как dpkg шаг за шагом регистрирует все операции, выполняемые для отдельных пакетов, aptitude предоставляет более широкий обзор операций высокого уровня, таких как общесистемное обновление.

Но будьте внимательны, так как данный журнал содержит только сводку операций, выполненных aptitude. Если используются другие интерфейсы (или даже dpkg), то журнал aptitude будет включать только частичный обзор операций, поэтому не стоит полагаться на него при желании получить достоверную историю системы.

Synaptic

Synaptic — графический менеджер пакетов, который имеет понятный и эффективный графический интерфейс, основанный на GTK + и GNOME (рис. 8.2). В него включено множество готовых к использованию фильтров, которые обеспечивают быстрый доступ к новым пакетам, а также установленным, обновляемым, устаревшим и т. д. При просмотре этих списков вы можете выбрать операции для применения к пакетам (установить, обновить, удалить, очистить); операции выполняются не сразу, а сначала заносятся в список задач. Одно нажатие кнопки подтверждает операции, и они сразу же выполняются.

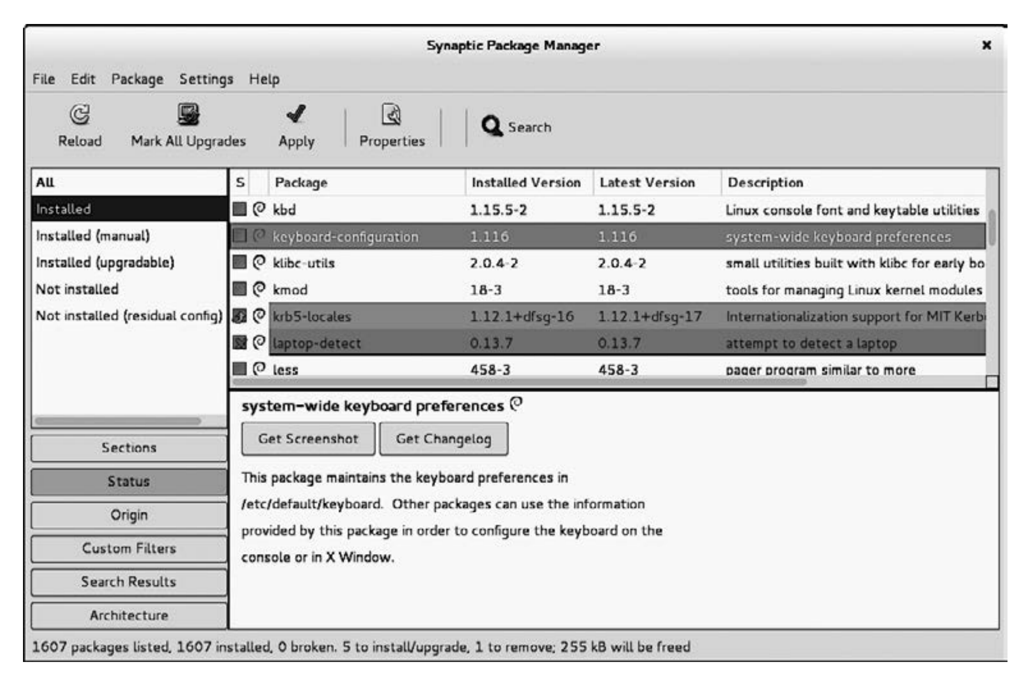


Рис. 8.2. Менеджер пакетов Synaptic

8.3. Дополнительная настройка и использование APT

Теперь пришло время погрузиться в более сложные темы. Для начала рассмотрим расширенную конфигурацию APT; она позволит установить больше постоянных параметров, которые будут применяться к инструментам APT. Затем покажем, как управлять приоритетами пакетов, что предоставит доступ к более настроенным, индивидуализированным обновлениям и усовершенствованиям. Мы также научим обращаться с несколькими дистрибутивами, чтобы вы могли начать экспериментировать с пакетами, поступающими из других дистрибутивов. Затем рассмотрим,

как отслеживать автоматически установленные пакеты, что позволит управлять пакетами, которые устанавливаются через зависимости. Мы также объясним, как многоархитектурная поддержка открывает возможность запуска пакетов, созданных для различных аппаратных архитектур.

И последнее, но не менее важное: мы обсудим криптографические протоколы и утилиты, которые позволят проверить подлинность любого пакета.

Настройка APT

Прежде чем погрузиться в настройку APT, рассмотрим механизм настройки системы Debian. Изначально настройка выполнялась с помощью специализированных файлов конфигурации. Однако в современных Linux-системах, таких как Debian и Kali, все чаще используются конфигурационные каталоги с расширением `.d`. Каждый каталог представляет собой файл конфигурации, который разбит на несколько файлов. В этом смысле все файлы в каталоге `/etc/apt/apt.conf.d/` являются инструкциями для настройки APT. Инструмент обрабатывает файлы в алфавитном порядке, так что более поздние могут изменять элементы конфигурации, определенные в более ранних файлах.

Данная структура привносит некоторую гибкость в деятельность администраторов и разработчиков пакетов, позволяя им изменять настройки программного обеспечения с помощью добавления файлов, не прибегая к необходимости изменять существующий файл. Это особенно полезно для создателей пакетов, поскольку они могут применять данный подход для адаптации конфигурации другого программного обеспечения с целью гарантировать, что оно отлично сосуществует с их ПО, не нарушая политику Debian, строго запрещающую изменять файлы конфигурации других пакетов. Благодаря механизму конфигурации с использованием расширения `.d` вам не нужно вручную выполнять множество инструкций конфигурации пакета, которые обычно находятся в файле пакета `/usr/share/doc/пакет/README.Debian`, так как программа установки может работать с файлами конфигурации.

Будьте осторожны с файлами конфигурации, созданными из каталогов .d

Хотя APT имеет встроенную поддержку своего каталога /etc/apt/apt.conf.d, это не всегда работает. Для некоторых приложений (например, like exim) каталог .d является дополнением для Debian, применяемым в качестве входных данных для динамического создания канонического файла конфигурации, используемого приложением. В таких случаях пакеты предоставляют команду update-* (например, update-exim4.conf), которая объединяет файлы из каталога .d и перезаписывает основной файл конфигурации.

В этом случае не стоит редактировать вручную основной файл конфигурации, так как ваши изменения будут потеряны при следующем выполнении команды update-*. Кроме того, вы не должны забывать запустить эту команду после редактирования файла из каталога .d (или ваши изменения не будут применены).

Разобравшись с понятием механизма конфигурации `.d`, поговорим о способах его использования для настройки APT. Мы уже говорили, что вы можете менять поведение APT с помощью параметров для `dpkg`, как в этом примере, который выполняет принудительную перезапись при установке пакета `zsh`:

```
# apt -o Dpkg::Options::="--force-overwrite" install zsh
```

Очевидно, это очень трудоемкий процесс, особенно если вы часто используете параметры, но вы можете применить структуру конфигурации каталога `.d`, чтобы настроить некоторые аспекты работы APT, добавив директивы в файл в каталоге `/etc/apt/apt.conf.d/`.

Скажем, директиву из примера ниже (как и любую другую) можно легко добавить в файл каталога `/etc/apt/apt.conf.d/`. Имя этого файла несколько условно, но общим соглашением является использование имен либо `local`, либо `99local`:

```
$ cat /etc/apt/apt.conf.d/99local
Dpkg::Options {
   "--force-overwrite";
}
```

Существует множество других полезных параметров конфигурации, и мы, конечно же, не можем охватить их все, но обратим внимание на один, связанный с подключением к сети. Например, если вы можете получить доступ к сети только через прокси, то добавьте следующую строку: `Acquire::http::proxy "http://`*`ваш_прокси`*`:3128"`. Для FTP-прокси используйте `Acquire::ftp::proxy "ftp://`*`ваш_прокси`*`"`.

Чтобы узнать о других параметрах конфигурации, ознакомьтесь с руководством `apt.conf(5)` с помощью команды `man apt.conf` (подробнее о страницах руководства см. в подразделе «Руководства»).

Управление приоритетами пакетов

Один из наиболее важных аспектов конфигурации APT — управление приоритетами, связанными с каждым источником пакета. Например, можно расширить систему Kali Rolling одним или двумя новыми пакетами из Debian Unstable или Debian Experimental. Можно назначить приоритет для каждого доступного пакета (один и тот же пакет может иметь несколько приоритетов в зависимости от его версии или дистрибутива, которым он предоставляется). Эти приоритеты будут влиять на поведение APT: для каждого пакета всегда выбирается версия с наивысшим приоритетом (за исключением случаев, когда эта версия старше установленной, а ее приоритет меньше 1000).

APT определяет несколько приоритетов по умолчанию. Каждая установленная версия пакета имеет приоритет 100. Неустановленная версия имеет приоритет 500 по умолчанию, но может подняться до 990, если является частью целевой версии (определенной с помощью параметра командной строки `-t` (от target — «цель») или директивы конфигурации `APT::Default-Release`).

Вы можете изменять приоритеты, добавляя записи в файл `/etc/apt/preferences` с именами интересующих вас пакетов, их версиями, источниками и новыми приоритетами.

APT не станет устанавливать более старую версию пакета (то есть пакет, номер версии которого меньше номера версии текущего пакета), за исключением случаев, когда его приоритет выше 1000. APT всегда будет устанавливать пакет с наивысшим приоритетом, следуя таким правилам: в ситуации, когда два пакета имеют одинаковые приоритеты, APT устанавливает более новый (номер версии которого выше). Если два пакета одинаковой версии имеют одинаковый приоритет, но отличаются содержимым, то APT устанавливает версию, которая не установлена (это правило было создано для случая обновления пакета без изменения номера версии, что, как правило, необходимо).

Пакет, приоритет которого меньше 0, никогда не будет установлен. Пакет с приоритетом в диапазоне от 0 до 100 будет установлен только в том случае, если никакая другая версия пакета еще не установлена. Пакет с приоритетом от 100 до 500 будет установлен, только если нет другой, более новой, версии (установленной или доступной в другом дистрибутиве). Пакет с приоритетом между 501 и 990 будет установлен только при отсутствии другой, более новой, версии (установленной или доступной в целевом дистрибутиве). Пакет с приоритетом между 990 и 1000 будет установлен, за исключением ситуации, когда установленная версия новее. Приоритет, превышающий 1000, всегда приведет к установке пакета, даже если это заставит APT установить более раннюю версию.

Когда APT просматривает `/etc/apt/preferences`, сначала учитываются наиболее специфические записи (часто указывающие на конкретный пакет), а затем более общие (которые касаются, например, всех пакетов дистрибутива). Если существует несколько общих записей, то используется первая подходящая. Доступные критерии выбора включают имя пакета и источник, предоставляющий его. Каждый источник пакета идентифицируется информацией, содержащейся в файле `Release`, который APT скачивает вместе с файлами `Packages`. Последние указывают источник, обычно Kali для пакетов с официальных зеркал Kali и Debian для пакетов с официальных зеркал Debian, но для сторонних репозиториев источником может быть и имя человека или организации. Файл `Release` также предоставляет имя дистрибутива вместе с его версией. Рассмотрим его синтаксис с помощью нескольких конкретных примеров этого механизма.

Приоритет Kali-Bleeding-Edge и Debian Experimental

Если в файле sources.list вы указали kali-bleeding-edge или Debian experimental, то соответствующие пакеты почти никогда не будут установлены, поскольку их приоритет APT по умолчанию равен 1. Так сделано специально для того, чтобы пользователи не устанавливали пакеты bleeding edge по ошибке. Пакеты могут устанавливаться только через выполнение команды apt install *пакет*/kali-bleeding-edge; при этом предполагается, конечно,

что вы осознаете все риски и потенциальную головную боль. Кроме того, возможно (хотя и не рекомендуется) обрабатывать пакеты kali-bleeding-edge/experimental подобно пакетам других дистрибутивов, предоставляя им приоритет 500. Это делается с помощью особой записи в /etc/apt/preferences:

```
Package: *
Pin: release a=kali-bleeding-edge
Pin-Priority: 500
```

Предположим, вы предпочитаете только пакеты из Kali и хотите, чтобы пакеты Debian устанавливались лишь при явном запросе. Для этого можете добавить следующие записи в файл `/etc/apt/preferences` (или в любой другой файл каталога `/etc/apt/preferences.d/`):

```
Package: *
Pin: release o=Kali
Pin-Priority: 900

Package: *
Pin: release o=Debian
Pin-Priority: -10
```

В двух последних примерах вы встречали строку `a=kali-bleeding-edge`, определяющую имя выбранного дистрибутива, и строки `o=Kali` и `o=Debian`, ограничивающие область применения пакетами, происхождение которых Kali и Debian соответственно. Предположим теперь, что у вас есть сервер с несколькими локальными программами, зависящими от версии Perl 5.22, и вы не хотите, чтобы обновления устанавливали другую версию. Вы можете использовать следующую запись:

```
Package: perl
Pin: version 5.22*
Pin-Priority: 1001
```

Справочная документация для этого файла конфигурации доступна на странице руководства `apt_preferences(5)`, которую можно вывести на экран с помощью команды `man apt_preferences`.

Добавление комментариев в файле /etc/apt/preferences

Официального синтаксиса для комментариев в /etc/apt/preferences нет, но можно добавить некоторые текстовые описания с помощью одного или нескольких полей Explanation (Объяснение) для любой записи:

```
Explanation: The package xserver-xorg-video-intel provided
Explanation: in experimental can be used safely
Package: xserver-xorg-video-intel
Pin: release a=experimental
Pin-Priority: 500
```

Работа с несколькими дистрибутивами

Учитывая все прелести инструмента `apt`, вы, скорее всего, захотите копнуть глубже и начать экспериментировать с пакетами, поступающими из других дистрибутивов. Например, после установки системы Kali Rolling вы, возможно, захотите попробовать установить пакеты, доступные в Kali Dev, Debian Unstable или Debian Experimental, при этом не слишком сильно изменяя систему относительно ее исходного состояния.

Даже если вы случайно столкнетесь с проблемами при совместном использовании пакетов из разных дистрибутивов, `apt` прекрасно справляется с таким сосуществованием и очень эффективно ограничивает риски (при условии, что зависимости пакетов точны). Для начала перечислите все дистрибутивы, применяемые в `/etc/apt/sources.list`, и определите исходный дистрибутив с помощью параметра `APT::Default-Release` (см. раздел «Обновление Kali Linux» раздела 8.2).

Предположим, что Kali Rolling — ваш исходный дистрибутив, но Kali Dev и Debian Unstable также присутствуют в вашем файле `sources.list`. В этом случае вы можете использовать команду `apt install` *`пакет`*`/unstable` для установки пакета из Debian Unstable. Если установка не выполняется из-за нарушения некоторых зависимостей, то позвольте разрешить эти зависимости с Unstable, добавив параметр `-t unstable`.

В таком случае обновления (`upgrade` и `full-upgrade`) выполняются в Kali Rolling, за исключением пакетов из других дистрибутивов: они будут отслеживать обновления, доступные в этих дистрибутивах. Мы объясним данное поведение ниже по тексту с помощью приоритетов, установленных APT по умолчанию. Смело применяйте команду `apt-cache policy` (см. врезку «Использование команды apt-cache policy» ниже) для проверки указанных приоритетов.

Все основано на факте, что APT рассматривает пакеты с более высокой или равной версией, чем версия установленного пакета (при условии, что `/etc/apt/preferences` не используется для поднятия приоритетов для некоторых пакетов выше 1000).

Использование команды apt-cache policy

Чтобы лучше понять механизм приоритетов, смело выполняйте команду apt-cache policy с целью узнать приоритет по умолчанию для каждого источника пакета. Вы также можете применить команду apt-cache policy пакет для отображения приоритетов всех доступных версий данного пакета.

Предположим, что вы установили версию 1 первого пакета из *Kali Rolling* и версии 2 и 3 доступны в *Kali Dev* и *Debian Unstable* соответственно. Установленная версия имеет приоритет 100, но версия, доступная в *Kali Rolling* (та же самая), имеет приоритет 990 (так как это часть целевого дистрибутива). Пакеты в *Kali Dev* и *Debian Unstable* имеют приоритет 500 (приоритет по умолчанию для неуста-

новленной версии). Таким образом, побеждает версия 1 с приоритетом 990. Пакет остается в *Kali Rolling*.

А теперь рассмотрим пример с другим пакетом, версия 2 которого была установлена из *Kali Dev*. Версия 1 доступна в *Kali Rolling*, а версия 3 — в *Debian Unstable*. Версия 1 (с приоритетом 990, то есть ниже 1000) отбрасывается, поскольку ниже установленной версии. Остаются версии 2 и 3, обе с приоритетом 500. Столкнувшись с такой альтернативой, APT выбирает самую новую версию, ту, что из *Debian Unstable*. Если вы не хотите, чтобы пакет, установленный из *Kali Dev*, был перенесен на *Debian Unstable*, то необходимо назначить приоритет ниже 500 (например 490) для пакетов, поступающих из *Debian Unstable*. Для этого можно изменить `/etc/apt/preferences` следующим образом:

```
Package: *
Pin: release a=unstable
Pin-Priority: 490
```

Отслеживание автоматически установленных пакетов

Одна из важнейших функций `apt` — отслеживание пакетов, установленных через зависимости. Эти пакеты называются *автоматическими* и часто включают библиотеки.

Когда пакеты удаляются, менеджеры пакетов могут составить список автоматических пакетов, которые больше не нужны (так как нет установленных вручную пакетов, зависящих от них). Команда `apt autoremove` избавится от этих пакетов.

У инструмента Aptitude нет такой команды, поскольку он автоматически удаляет подобные пакеты сразу после их идентификации. Во всех случаях инструменты выводят сообщение с перечнем затронутых пакетов.

Полезной привычкой станет отмечать как автоматический любой пакет, непосредственно в котором нет нужды, чтобы эти пакеты удалялись автоматически. Для этого можно использовать команду `apt-mark auto` *`пакет`*, тогда как команда `apt-mark manual` *`пакет`* выполняет обратное действие. Команды `aptitude markauto` и `aptitude unmarkauto` работают аналогично, при этом предлагают больше возможностей для маркировки сразу нескольких пакетов (см. пункт «Aptitude» подраздела «Пользовательские интерфейсы: aptitude и synaptic» раздела 8.2). Интерактивный интерфейс `aptitude` также позволяет с легкостью проверить наличие автоматического флага для нескольких пакетов сразу.

Возможно, вам захочется узнать, почему в системе присутствует автоматически установленный пакет. Чтобы получить эту информацию с помощью командной строки, воспользуйтесь командой `aptitude why` *`пакет`* (`apt` и `apt-get` не имеют подобной функции):

```
$ aptitude why python-debian
i aptitude           Recommends apt-xapian-index
i A apt-xapian-index Depends    python-debian (>= 0.1.15)
```

Использование поддержки Multi-Arch

Все пакеты Debian содержат поле `Architecture` (Архитектура) в своих данных управления. Это поле может содержать либо значение `all` (все) (для не зависящих от архитектуры пакетов), либо имя архитектуры, для которой он предназначен (например, amd64 или armhf). В последнем случае `dpkg` по умолчанию установит пакет, только если его архитектура соответствует архитектуре системы, что можно проверить с помощью команды `dpkg --print-architecture`.

Это ограничение гарантирует, что вы не получите в результате двоичные файлы, скомпилированные для неправильной архитектуры. Все было бы идеально, за исключением одного обстоятельства: (некоторые) компьютеры могут запускать двоичные файлы для нескольких архитектур либо самостоятельно (система amd64 может запускать бинарные файлы i386), либо через эмуляторы.

Подключение Multi-Arch

Многоархитектурная поддержка `dpkg` позволяет пользователям определять внешние архитектуры, которые могут быть установлены в данной системе. Это легко сделать с помощью команды `dpkg --add-architecture`, как в примере ниже, где архитектура i386 должна быть добавлена в систему amd64, чтобы запускать приложения Windows, применяя Wine (https://www.winehq.org/). Есть соответствующая команда `dpkg --remove-architecture` для сброса поддержки внешней архитектуры, но ее можно использовать, только когда не остается установленных пакетов данной архитектуры.

```
# dpkg --print-architecture
amd64
# wine
it looks like wine32 is missing, you should install it.
multiarch needs to be enabled first. as root, please
execute "dpkg --add-architecture i386 & apt-get update &
apt-get install wine32"
Usage: wine PROGRAM [ARGUMENTS...]   Run the specified program
       wine --help                   Display this help and exit
       wine --version                Output version information and exit
# dpkg --add-architecture i386
# dpkg --print-foreign-architectures
i386
# apt update
[...]
# apt install wine32
[...]
Setting up libwine:i386 (1.8.6-5) ...
Setting up vdpau-driver-all:i386 (1.1.1-6) ...
Setting up wine32:i386 (1.8.6-5) ...
Setting up libasound2-plugins:i386 (1.1.1-1) ...
Processing triggers for libc-bin (2.24-9)
```

```
# wine
Usage: wine PROGRAM [ARGUMENTS...]    Run the specified program
       wine --help                    Display this help and exit
       wine --version                 Output version information and exit
# dpkg --remove-architecture i386
dpkg: error: cannot remove architecture 'i386' currently in use by the
database
# dpkg --print-foreign-architectures
i386
```

APT автоматически обнаружит, когда `dpkg` настроен для поддержки внешних архитектур, и начнет скачивание соответствующих файлов `Packages` в процессе обновления.

Затем можно установить внешние пакеты с помощью команды `apt install` *`пакет`*`:architecture`.

Использование проприетарных бинарных файлов i386 в системе amd64

Существует много вариантов применения мультиархитектурности, но наиболее популярный — возможность выполнения 32-разрядных бинарных файлов (i386) в 64-разрядных системах (amd64), особенно учитывая, что некоторые популярные проприетарные приложения (например, Skype) предоставляются только в 32-разрядных версиях.

Изменения, связанные с Multi-Arch

Чтобы сделать мультиархитектурность действительно полезной и удобной для использования, библиотеки пришлось переупаковать и переместить в специализированный для архитектуры каталог, позволяя устанавливать несколько копий пакетов (предназначенных для различных архитектур) одновременно. Такие обновленные пакеты содержат `Multi-Arch`: у них одинаковые заголовки; это говорит системе пакетирования о возможности безопасно совместно установить различные архитектуры пакета (и что эти пакеты могут удовлетворять только зависимостям пакетов той же архитектуры).

```
$ dpkg -s libwine
dpkg-query: error: --status needs a valid package name but 'libwine' is not:
    ➥ambiguous package name 'libwine' with more than one installed instance

Use --help for help about querying packages.
$ dpkg -s libwine:amd64 libwine:i386 | grep ^Multi
Multi-Arch: same
Multi-Arch: same
$ dpkg -L libgcc1:amd64 |grep .so
[...]
/usr/lib/x86_64-linux-gnu/wine/libwine.so.1
$ dpkg -S /usr/share/doc/libwine/copyright
libwine:amd64, libwine:i386: /usr/share/doc/libwine/copyright
```

Стоит отметить: в именах пакетов `Multi-Arch: same` должно быть указано, к какой архитектуре они относятся, чтобы их можно было легко идентифицировать. Они могут совместно использовать файлы с другими экземплярами одного и того же пакета; `dpkg` гарантирует, что все пакеты применяют идентичные файлы с точностью до бита, когда эти файлы являются общими. Кроме того, все экземпляры пакета должны иметь одну и ту же версию и обновляться вместе.

Поддержка Multi-Arch также привносит ряд трудностей в процесс обработки зависимостей. Для удовлетворения зависимости нужен либо пакет с отметкой `Multi-Arch: foreign`, либо пакет, архитектура которого соответствует одному из пакетов, объявляющих зависимость (в процессе разрешения зависимостей предполагается, что независимые от архитектуры пакеты имеют такую же архитектуру, как и у системы). Кроме того, зависимость может быть ослаблена, чтобы позволить любой архитектуре выполнить ее, с помощью синтаксиса *пакет*`:any`, но внешние пакеты могут удовлетворять такую зависимость только при наличии отметки `Multi-Arch: allowed`.

Проверка подлинности пакета

Обновление системы — очень чувствительная операция, и крайне важно устанавливать только официальные пакеты из хранилищ Kali. Если зеркало Kali, которое вы используете, было взломано, то злоумышленник мог добавить вредоносный код в официальный пакет. После установки такого пакета он сможет выполнить все задуманное взломщиком, включая раскрытие паролей или конфиденциальной информации. Чтобы обойти этот риск, Kali использует защитные цифровые пломбы от несанкционированного доступа с целью гарантировать, что на момент установки пакет действительно исходит от его официального разработчика и не был изменен третьим лицом.

Пломба работает с помощью криптографических хешей и подписи. Подписанный файл — `Release`, предоставляемый зеркалами Kali. Он содержит список файлов `Packages` (включая их сжатые формы, `Packages.gz` и `Packages.xz`, и дополнительные версии), а также их хеши MD5, SHA1 и SHA256, которые гарантируют, что файлы не были подделаны. Эти файлы `Packages` включают список пакетов Debian, доступных на зеркале, вместе с их хешами; это, в свою очередь, гарантирует отсутствие изменений в содержимом самих пакетов.

Управляет доверенными ключами команда `apt-key`, присутствующая в пакете `apt`. Данная программа обслуживает связку открытых ключей GnuPG, которые используются для проверки подписей в файлах `Release.gpg`, доступных на зеркалах. Ее можно применять для добавления новых ключей вручную (когда необходимо задействовать неофициальные зеркала). Но, как правило, нужны только официальные ключи Kali. Эти ключи автоматически обновляются с помощью пакета `kali-archive-keyring` (который помещает соответствующие связки ключей в `/etc/apt/trusted.gpg.d`). Однако первая установка данного пакета

требует осторожности: даже если пакет подписан, как и все остальные, подпись нельзя проверить извне. Поэтому осторожные администраторы должны проверять цифровые подписи импортированных ключей, прежде чем доверить им установку новых пакетов:

```
# apt-key fingerprint
/etc/apt/trusted.gpg.d/debian-archive-jessie-automatic.gpg
-----------------------------------------------------------
pub   4096R/2B90D010 2014-11-21 [expires: 2022-11-19]
Key   fingerprint = 126C 0D24 BD8A 2942 CC7D F8AC 7638 D044 2B90 D010
uid              Debian Archive Automatic Signing Key (8/jessie)
                 ➥<ftpmaster@debian.org>

/etc/apt/trusted.gpg.d/debian-archive-jessie-security-automatic.gpg
--------------------------------------------------------------------
pub   4096R/C857C906 2014-11-21 [expires: 2022-11-19]
Key   fingerprint = D211 6914 1CEC D440 F2EB 8DDA 9D6D 8F6B C857 C906
uid              Debian Security Archive Automatic Signing Key (8/jessie)
                 ➥<ftpmaster@debian.org>

/etc/apt/trusted.gpg.d/debian-archive-jessie-stable.gpg
--------------------------------------------------------
pub   4096R/518E17E1 2013-08-17 [expires: 2021-08-15]
Key   fingerprint = 75DD C3C4 A499 F1A1 8CB5 F3C8 CBF8 D6FD 518E 17E1
uid              Jessie Stable Release Key <debian-release@lists.debian.org>

/etc/apt/trusted.gpg.d/debian-archive-squeeze-automatic.gpg
------------------------------------------------------------
pub   4096R/473041FA 2010-08-27 [expires: 2018-03-05]
Key   fingerprint = 9FED 2BCB DCD2 9CDF 7626 78CB AED4 B06F 4730 41FA
uid              Debian Archive Automatic Signing Key (6.0/squeeze)
                 ➥<ftpmaster@debian.org>

/etc/apt/trusted.gpg.d/debian-archive-squeeze-stable.gpg
---------------------------------------------------------
pub   4096R/B98321F9 2010-08-07 [expires: 2017-08-05]
Key   fingerprint = 0E4E DE2C 7F3E 1FC0 D033 800E 6448 1591 B983 21F9
uid              Squeeze Stable Release Key <debian-release@lists.debian.org>

/etc/apt/trusted.gpg.d/debian-archive-wheezy-automatic.gpg
-----------------------------------------------------------
pub   4096R/46925553 2012-04-27 [expires: 2020-04-25]
Key   fingerprint = A1BD 8E9D 78F7 FE5C 3E65 D8AF 8B48 AD62 4692 5553
uid              Debian Archive Automatic Signing Key (7.0/wheezy)
                 ➥<ftpmaster@debian.org>

/etc/apt/trusted.gpg.d/debian-archive-wheezy-stable.gpg
--------------------------------------------------------
pub   4096R/65FFB764 2012-05-08 [expires: 2019-05-07]
```

```
Key   fingerprint = ED6D 6527 1AAC F0FF 15D1 2303 6FB2 A1C2 65FF B764
uid              Wheezy Stable Release Key <debian-release@lists.debian.org>

/etc/apt/trusted.gpg.d/kali-archive-keyring.gpg
-----------------------------------------------
pub   4096R/7D8D0BF6 2012-03-05 [expires: 2018-02-02]
Key   fingerprint = 44C6 513A 8E4F B3D3 0875 F758 ED44 4FF0 7D8D 0BF6
uid              Kali Linux Repository <devel@kali.org>
sub   4096R/FC0D0DCB 2012-03-05 [expires: 2018-02-02]
```

Когда в файл `sources.list` добавляется сторонний источник пакетов, необходимо сообщить APT, что соответствующему проверочному ключу GPG стоит доверять (в противном случае программа будет постоянно «жаловаться», что не может гарантировать подлинность пакетов, поступающих из этого репозитория). Первым шагом, конечно, является получение открытого ключа. Чаще всего последний предоставляется в виде небольшого текстового файла, который в последующих примерах мы будем называть `key.asc`.

Чтобы добавить ключ к связке доверенных ключей, администратор может выполнить команду `apt-key add <key.asc`. Кроме того, можно использовать графический интерфейс Synaptic: его вкладка Authentication (Аутентификация) в меню Settings ▸ Repositories (Настройки ▸ Хранилища) позволяет импортировать ключ из файла `key.asc`.

Те, кто предпочитает применять специализированное приложение и получать более подробную информацию о доверенных ключах, могут воспользоваться `gui-apt-key` (в одноименном пакете) — небольшим графическим интерфейсом, который управляет связкой доверенных ключей.

После того как соответствующие ключи добавлены в связку, APT проверяет подписи перед выполнением любой рискованной операции. Таким образом программа выдаст предупреждение при попытке установить пакет, подлинность которого невозможно подтвердить.

8.4. Справка по пакетам: погружение в систему пакетов Debian

Пришло время углубиться в систему пакетов Debian и Kali. В данном разделе мы немного выйдем за рамки инструментов и синтаксиса и сосредоточимся на так называемых «гайках и болтах» упаковочной системы. Этот взгляд «за кулисы» поможет понять работу APT начиная с его основ и даст представление о том, как значительно упорядочить и настроить вашу систему Kali. Возможно, вы не запомните абсолютно все, о чем здесь будет сказано, однако рассматриваемый и справочный материалы, безусловно, сослужат хорошую службу для вашего продвижения в освоении системы Kali Linux.

До сих пор вы взаимодействовали с данными пакета APT с помощью различных инструментов, предназначенных для работы с ним. Теперь мы пойдем дальше и заглянем внутрь пакетов, рассмотрим внутреннюю метаинформацию (или ин-

формацию о другой информации), которую используют инструменты управления пакетами.

Это сочетание файлового архива и метаинформации можно увидеть непосредственно в структуре файла формата `.deb`, который является простым архивом `ar`, объединяющим три файла:

```
$ ar t /var/cache/apt/archives/apt_1.4~beta1_amd64.deb
debian-binary
control.tar.gz
data.tar.xz
```

Файл `debian-binary` содержит лишь номер версии, описывающий формат архива:

```
$ ar p /var/cache/apt/archives/apt_1.4~beta1_amd64.deb debian-binary
2.0
```

В архиве `control.tar.gz` находится метаинформация:

```
$ ar p /var/cache/apt/archives/apt_1.4~beta1_amd64.deb control.tar.gz | tar -tzf -
./
./conffiles
./control
./md5sums
./postinst
./postrm
./preinst
./prerm
./shlibs
./triggers
```

И наконец, архив `data.tar.xz` (формат сжатия может отличаться) включает сами файлы, которые необходимо установить в файловой системе:

```
$ ar p /var/cache/apt/archives/apt_1.4~beta1_amd64.deb data.tar.xz | tar -tJf -
./
./etc/
./etc/apt/
./etc/apt/apt.conf.d/
./etc/apt/apt.conf.d/01autoremove
./etc/apt/preferences.d/
./etc/apt/sources.list.d/
./etc/apt/trusted.gpg.d/
./etc/cron.daily/
./etc/cron.daily/apt-compat
./etc/kernel/
./etc/kernel/postinst.d/
./etc/kernel/postinst.d/apt-auto-removal
./etc/logrotate.d/
./etc/logrotate.d/apt
./lib/
./lib/systemd/
[...]
```

Обратите внимание: в этом примере рассматривается пакет `.deb` в кэше архива APT, и ваш архив может включать файлы с номерами версий, отличными от показанного.

В данном разделе мы представим метаинформацию, содержащуюся в каждом пакете, и покажем, как ее использовать.

Файл control

Начнем с файла `control`, который находится в архиве `control.tar.gz`. Файл содержит всю самую важную информацию о пакете. Он использует структуру, похожую на заголовки электронной почты, и его можно просмотреть с помощью команды `dpkg -I`. Например, файл `control` для `apt` выглядит так:

```
$ dpkg -I apt_1.4~beta1_amd64.deb control
Package: apt
Version: 1.4~beta1
Architecture: amd64
Maintainer: APT Development Team <deity@lists.debian.org>
Installed-Size: 3478
Depends: adduser, gpgv | gpgv2 | gpgv1, debian-archive-keyring, init-system-
    ➥helpers (>= 1.18~), libapt-pkg5.0 (>= 1.3~rc2), libc6 (>= 2.15),
    ➥libgcc1 (>= 1:3.0), libstdc++6 (>= 5.2)
Recommends: gnupg | gnupg2 | gnupg1
Suggests: apt-doc, aptitude | synaptic | wajig, dpkg-dev (>= 1.17.2),
    ➥powermgmt-base, python-apt
Breaks: apt-utils (<< 1.3~exp2~)
Replaces: apt-utils (<< 1.3~exp2~)
Section: admin
Priority: important
Description: commandline package manager
 This package provides commandline tools for searching and
 managing as well as querying information about packages
 as a low-level access to all features of the libapt-pkg library.
 .
 These include:
  * apt-get for retrieval of packages and information about them
    from authenticated sources and for installation, upgrade and
    removal of packages together with their dependencies
  * apt-cache for querying available information about installed
    as well as installable packages
  * apt-cdrom to use removable media as a source for packages
  * apt-config as an interface to the configuration settings
  * apt-key as an interface to manage authentication keys
```

В этом подразделе мы разберем файл `control` подробнее и объясним значение его различных полей. Каждое из них даст лучшее представление о системе пакетирования, позволит точнее настроить управление конфигурацией и обеспечит вас информацией, необходимой для устранения потенциальных проблем.

Зависимости: поле Depends

Зависимости пакетов определяются в поле `Depends` в заголовке пакета. Это список условий для корректной работы пакета — данная информация используется такими инструментами, как `apt`, чтобы установить правильные версии необходимых библиотек, которые удовлетворяют зависимости устанавливаемого пакета. Для каждой зависимости диапазон версий, соответствующих данному условию, может быть ограничен. Другими словами, можно сказать, что нам требуется пакет `libc6` версии не ниже 2.15 (указывается `libc6 (>= 2.15)`). Операторы сравнения версий следующие:

- << — меньше;
- <= — меньше или равна;
- = — равна (однако `2.6.1` — не то же самое, что `2.6.1-1`);
- >= — больше или равна;
- >> — больше.

В списке условий запятая играет роль разделителя. Ее следует интерпретировать как логическое И. Внутри условий вертикальная черта (`|`) означает логическое ИЛИ (включающее ИЛИ, а не исключающее «строго одно из»). Поскольку оно имеет более высокий приоритет, чем И, то его можно использовать нужное количество раз. Так, зависимость «(A или B) и C» записывается в виде `A | B, C`. Напротив, выражение «A или (B и C)» следует записывать как «(A или B) и (A или C)», так как поле `Depends` не допускает использования скобок, меняющих порядок приоритетов между логическими операторами ИЛИ и И. То есть нужно писать `A | B, A | C`. Дополнительную информацию можно найти на странице https://www.debian.org/doc/debian-policy/#document-ch-relationships.

Система зависимостей — хороший механизм для поддержания работоспособности программ, но у нее имеется и другое применение — «метапакеты». Это пустые пакеты, в которых описаны только зависимости. Они обеспечивают установку группы взаимосвязанных программ, выбранных создателем метапакета; соответственно, команда `apt install` *`метапакет`* автоматически установит все эти программы, используя зависимости метапакета. Пакеты `gnome`, `kde-full` и `linux-image-amd64` являются примерами метапакетов.

Pre-Depends — более требовательные зависимости

Предварительные зависимости, перечисленные в поле `Pre-Depends` заголовков пакета, дополняют обычные зависимости; их синтаксис аналогичен. Обычная зависимость показывает, что пакет должен быть распакован и настроен до настройки зависимого пакета. Предварительная зависимость оговаривает, что пакет должен быть распакован и настроен до запуска предустановочного сценария

пакета, для которого указана предварительная зависимость, то есть до его установки.

Предварительная зависимость очень требовательна к apt, поскольку добавляет строгие ограничения на порядок установки пакетов. Поэтому использование предварительных зависимостей без крайней необходимости не поощряется. Более того, перед добавлением предварительной зависимости рекомендовано проконсультироваться с другими разработчиками по адресу debian-devel@lists.debian.org. Как правило, удается найти другое решение или обходной путь.

Поля Recommends, Suggests и Enhances

В полях Recommends и Suggests указываются зависимости, не являющиеся обязательными. Рекомендуемые зависимости, наиболее важные, значительно улучшают функциональность, предоставляемую пакетом, но не выступают крайне необходимыми для его работы. Предлагаемые зависимости, следующие по значимости, означают, что некоторые пакеты могут дополнить устанавливаемый или быть полезными в связке с ним, но вполне целесообразной будет и установка одного без других.

Следует всегда устанавливать рекомендуемые пакеты, за исключением случаев, когда вы точно знаете, почему они не нужны. И наоборот, нет смысла устанавливать предлагаемые пакеты, если не знаете, зачем они вам.

В поле Enhances также указывается предложение, но другого рода. Оно на самом деле находится в предлагаемом пакете, а не в пакете, который выиграет от такого предложения. Смысл в том, что становится возможным добавить предложение, не меняя затрагиваемый пакет. Так, все дополнения, плагины и прочие расширения программы смогут появиться в списке предложений, относящихся к программе. Хотя оно существует уже несколько лет, это поле до сих пор по большей части игнорируется такими программами, как apt и synaptic. Смысл в том, чтобы предложения, вносимые через поле Enhances, отображались пользователю помимо обычных предложений — тех, которые находятся в поле Suggests.

Конфликты: поле Conflicts

Данное поле указывает на то, что пакет не может быть установлен, так как установлен другой пакет. Наиболее распространенные причины для этого — оба пакета включают файлы с одинаковыми именами, или сервисы предоставлены на одном и том же порту TCP, или файлы мешают работе друг друга.

Инструмент dpkg откажется установить пакет, если тот может вызвать конфликт с уже установленным пакетом, за исключением случаев, когда новый пакет указывает, что будет «заменять» установленный пакет, — тогда dpkg заменит старый пакет на новый. APT всегда следует вашим указаниям: если вы выберете установку нового пакета, то он автоматически предложит удалить проблемный пакет.

Несовместимость: поле Breaks

По своему действию поле `Breaks` похоже на поле `Conflicts`, но с особым значением. Оно сообщает, что установка пакета «сломает» другой пакет (или конкретные его версии). Как правило, такая несовместимость между пакетами имеет временный характер, и `Breaks` указывает на определенные несовместимые версии.

Инструмент `dpkg` откажется установить пакет, потенциально способный сломать уже установленный пакет, и `apt` попытается решить проблему, обновив пакет, которому грозит поломка, до более новой версии (которая, как предполагается, будет исправленной и, таким образом, снова совместимой).

Подобные ситуации могут возникнуть в случае обновления без обратной совместимости: это происходит, если новая версия работает не так, как старая, что приводит к сбою в другой программе при отсутствии должных мер. Поле `Breaks` помогает пользователю не сталкиваться с такими проблемами.

Предоставляемое пакетом: поле Provides

Это поле вводит очень интересное понятие «виртуального пакета». Оно имеет много применений, два из которых особенно важны. Первое состоит в использовании виртуального пакета с целью привязать к нему общий сервис (пакет предоставляет сервис). Второе показывает, что пакет полностью заменяет другой и при этом способен удовлетворять зависимости, которые удовлетворил бы заменяемый пакет. Таким образом, можно создать замену пакета, не прибегая к необходимости задействовать то же самое имя пакета.

Метапакет и виртуальный пакет

Очень важно четко понимать различие между метапакетами и виртуальными пакетами. Первые являются настоящими пакетами (то есть файлами формата .deb), единственное назначение которых состоит в выражении зависимостей.

Виртуальные пакеты, напротив, не существуют физически; они выступают только средством идентификации реальных пакетов на основании общих логических критериев (предоставляемого сервиса, совместимости со стандартной программой или ранее созданным пакетом и т. д.).

Предоставление сервиса. Рассмотрим первый случай более подробно на примере: все почтовые серверы, такие как *postfix* или *sendmail*, предоставляют виртуальный пакет `mail-transport-agent`. Поэтому в любом пакете, для работы которого нужен этот сервис (допустим, менеджер списков рассылки наподобие `smartlist` или `sympa`), просто указывается зависимость от `mail-transport-agent`, вместо того чтобы приводить большой и при этом все равно неполный список возможных решений. Кроме того, бесполезно устанавливать два почтовых сервера на одной машине, так что каждый из этих пакетов сообщит о конфликте

с виртуальным пакетом `mail-transport-agent`. Конфликт пакета с самим собой игнорируется системой, но данная технология не допустит установки двух почтовых серверов одновременно.

Взаимозаменяемость другим пакетом. Поле `Provides` также полезно в случаях, когда содержание пакета включается в состав другого, более крупного пакета. Например, модуль Perl `libdigest-md5-perl` был необязательным в Perl 5.6, но стал стандартным в Perl 5.8. Поэтому в пакете *`perl`* начиная с версии 5.8 указывается `Provides: libdigest-md5-perl`, чтобы зависимости от этого пакета были удовлетворены при установке Perl версии 5.8 (или новее). Сам пакет `libdigest-md5-perl` в конечном итоге подвергся удалению, поскольку после удаления старых версий Perl утратил смысл (рис. 8.3).

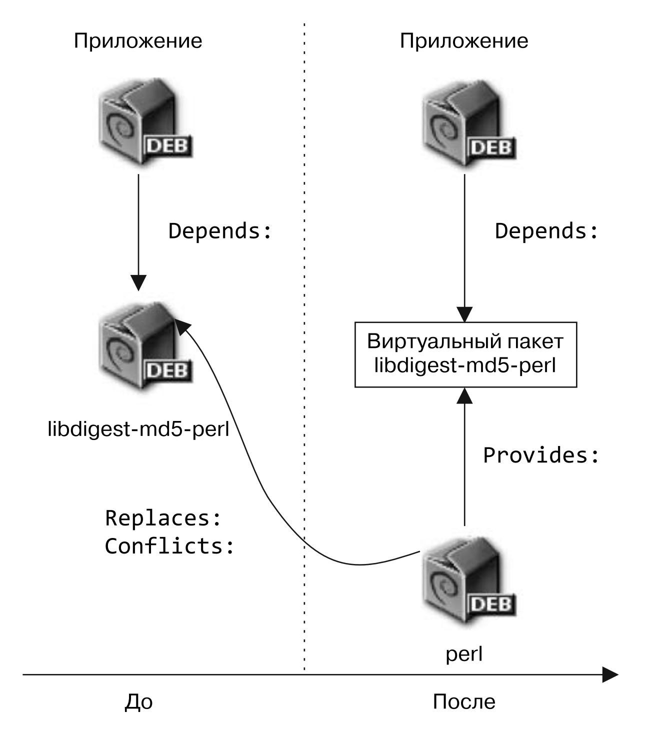


Рис. 8.3. Использование поля Provides во избежание нарушения зависимостей

Эта функция очень полезна, поскольку никогда нельзя предвидеть превратности процесса разработки, и важно иметь возможность подстроиться к переименованию устаревшего ПО или другим автоматическим заменам.

Замена файлов: поле Replaces

Данное поле указывает, что пакет содержит файлы, которые также присутствуют в другом пакете, но последний притом имеет право заменить их. Без этого поля

`dpkg` завершится с ошибкой, сообщив, что не может перезаписать файлы другого пакета (технически можно заставить его выполнить указанное действие с помощью параметра `--force-overwrite`, но это не является обоснованной стандартной операцией). Поле `Replaces` позволяет выявить потенциальные проблемы, но прежде чем добавлять его, разработчику нужно глубоко изучить данную тему.

Поле `Replaces` используется, когда у пакета меняется имя или один пакет включается в состав другого. Это также происходит в случае, если разработчик решает по-другому распределить файлы между двоичными пакетами, полученными из одного и того же исходного: замененный файл больше не принадлежит старому пакету, а только новому.

Если все файлы в установленном пакете были заменены, то принимается решение об удалении пакета. Это поле также указывает `dpkg` удалить замененный пакет в случае конфликта.

Сценарии конфигурации

Кроме файла `control` архив `control.tar.gz` в каждом пакете Debian может содержать несколько сценариев (`postinst`, `postrm`, `preinst`, `prerm`), которые `dpkg` вызывает на разных этапах обработки пакета. Поскольку эти файлы находятся в архиве пакета `.deb`, то для их отображения можно использовать команду `dpkg -I`:

```
$ dpkg -I /var/cache/apt/archives/zsh_5.3-1_amd64.deb | head
 new debian package, version 2.0.
 size 814486 bytes: control archive=2557 bytes.
     838 bytes,    20 lines      control
    3327 bytes,    43 lines      md5sums
     969 bytes,    41 lines    * postinst             #!/bin/sh
     348 bytes,    20 lines    * postrm               #!/bin/sh
     175 bytes,     5 lines    * preinst              #!/bin/sh
     175 bytes,     5 lines    * prerm                #!/bin/sh
 Package: zsh
 Version: 5.3-1
$ dpkg -I zsh_5.3-1_amd64.deb preinst
#!/bin/sh
set -e
# Automatically added by dh_installdeb
dpkg-maintscript-helper symlink_to_dir /usr/share/doc/zsh zsh-common
     ➥5.0.7-3 -- "$@"
# End automatically added section
```

В политике Debian подробно описаны все возможные случаи, в которых вызываются сценарии, и какие аргументы они притом получают. Эти последовательности могут быть довольно сложными, поскольку если один из сценариев завершается с ошибкой, то `dpkg` попытается вернуться к нормальному состоянию (насколько возможно), отменяя текущую установку или удаление.

База данных dpkg

Вы можете найти базу данных dpkg в файловой системе в каталоге /var/lib/dpkg/. Он содержит текущую запись всех пакетов, установленных в системе. Все сценарии конфигурации для установленных пакетов хранятся в каталоге /var/lib/dpkg/info/ в виде файла, префикс имени которого совпадает с именем пакета:

```
$ ls /var/lib/dpkg/info/zsh.*
/var/lib/dpkg/info/zsh.list
/var/lib/dpkg/info/zsh.md5sums
/var/lib/dpkg/info/zsh.postinst
/var/lib/dpkg/info/zsh.postrm
/var/lib/dpkg/info/zsh.preinst
/var/lib/dpkg/info/zsh.prerm
```

Этот каталог также включает файл с расширением .list для каждого пакета, содержащий список файлов, принадлежащих данному пакету:

```
$ head /var/lib/dpkg/info/zsh.list
/.
/bin
/bin/zsh
/bin/zsh5
/usr
/usr/lib
/usr/lib/x86_64-linux-gnu
/usr/lib/x86_64-linux-gnu/zsh
/usr/lib/x86_64-linux-gnu/zsh/5.2
/usr/lib/x86_64-linux-gnu/zsh/5.2/zsh
[...]
```

Файл /var/lib/dpkg/status содержит последовательности блоков данных (в формате небезызвестных почтовых заголовков, RFC 2822) с описанием статуса каждого пакета. Информация из файла control установленного пакета также дублируется сюда.

```
$ more /var/lib/dpkg/status
Package: gnome-characters
Status: install ok installed
Priority: optional
Section: gnome
Installed-Size: 1785
Maintainer: Debian GNOME Maintainers <pkg-gnome-maintainers@
    ➥lists.alioth.debian.org>
Architecture: amd64
Version: 3.20.1-1
[...]
```

Обсудим файлы конфигурации и посмотрим, как они взаимодействуют. Если вкратце, то сценарий `preinst` вызывается перед установкой пакета, а `postinst` — после нее. Аналогично `prerm` запускается перед удалением пакета, а `postrm` — после.

Обновление пакета эквивалентно удалению предыдущей версии и установке более новой. Все возможные ситуации описать здесь не получится, но мы рассмотрим две, встречающиеся чаще всего: установку/обновление и удаление.

Эти последовательности могут быть довольно запутанными, но визуальное представление способно помочь. Манож Сривастава нарисовал диаграммы, иллюстрирующие вызов конфигурационных сценариев `dpkg` (https://people.debian.org/~srivasta/MaintainerScripts.html). Похожие диаграммы также были разработаны проектом Debian Women (https://wiki.debian.org/MaintainerScripts); они несколько проще для понимания, но менее полные.

> Внимание!
> **Символьные имена скриптов**
>
> В последовательностях, описанных в этом подразделе, сценарии вызываются особыми именами, такими как old-prerm или new-postinst. Это, соответственно, сценарий prerm, содержавшийся в старой версии пакета (установленной до обновления), и сценарий postinst, входящий в новую версию (установленную при обновлении).

Последовательность сценариев установки и обновления

Вот что происходит во время установки пакета (или его обновления).

1. Чтобы выполнить обновление, `dpkg` выполняет команду `old-prerm upgrade` *`новая-версия`*.
2. Кроме того, для обновления `dpkg` выполняет команду `new-preinst upgrade` *`старая-версия`*; при установке работает команда `new-preinst install`. Последним параметром может быть добавлена старая версия, если пакет уже устанавливался раньше, однако был удален (но не вычищен, то есть конфигурационные файлы сохранились).
3. После этого распаковываются файлы нового пакета. Если файл уже существует, то заменяется, но создается его временная резервная копия.
4. При обновлении `dpkg` выполняется команда `old-postrm upgrade` *`новая-версия`*.
5. Инструмент `dpkg` обновляет все внутренние данные (список файлов, сценарии конфигурации и т. п.) и удаляет резервные копии замененных файлов. Теперь обратного пути нет: `dpkg` более недоступны все элементы, необходимые для отката к предыдущему состоянию.
6. Инструмент `dpkg` обновит все конфигурационные файлы, выводя запрос пользователю, если это невозможно сделать автоматически. Подробности этой процедуры рассмотрены в подразделе «Контрольные суммы, конфигурационные файлы» текущего раздела.
7. Наконец, `dpkg` настраивает пакет, выполняя команду `new-postinst configure` *`последняя-настроенная-версия`*.

Удаление пакета

Вот что происходит во время удаления пакета.

1. Инструмент `dpkg` выполняет команду `prerm remove`.
2. Инструмент `dpkg` удаляет все файлы пакета, за исключением конфигурационных файлов и сценариев конфигурации.
3. Инструмент `dpkg` выполняет команду `postrm remove`. Все сценарии настройки, за исключением `postrm`, удаляются. Если пользователь не использует опцию очистки, то процесс удаления заканчивается на этом шаге.
4. Для полного удаления пакета (в случае использования команды `dpkg --purge` или `dpkg -P`) также удаляются конфигурационные файлы и их копии (`*.dpkg-tmp`, `*.dpkg-old`, `*.dpkg-new`) и временные файлы; после этого `dpkg` выполняет команду `postrm purge`.

Четыре сценария, описанных выше, дополняются сценарием `config`, который предоставляют пакеты, применяющие `debconf`, чтобы запросить у пользователя информацию для настройки. Данный сценарий определяет вопросы, которые будут заданы `debconf` во время установки. Ответы заносятся в базу данных `debconf` для дальнейшего использования. Эти сценарии `apt` обычно выполняет до установки пакетов, последовательно, чтобы сгруппировать вопросы и задать их пользователю в начале процесса. Пред- и послеустановочные сценарии могут в дальнейшем применять эту информацию, чтобы действовать в соответствии с пожеланиями пользователей.

Инструмент debconf

Инструмент debconf был создан для решения постоянно повторявшейся в Debian проблемы. Все пакеты Debian, которые не могли работать без минимальной настройки, задавали вопросы, вызывая команды echo и read в послеустановочных сценариях postinst (и других похожих). Но это значило следующее: во время большой установки или обновления пользователь должен был находиться у компьютера, чтобы отвечать на различные вопросы, которые могли возникнуть в любое время. Необходимость в таких ручных вмешательствах теперь почти полностью отпала благодаря инструменту debconf.

У debconf множество интересных возможностей: взаимодействие с пользователем задается разработчиком; возможна локализация всех строк, отображаемых пользователю (все переводы хранятся в файле templates, описывающем взаимодействия); у него есть несколько интерфейсов (для текстового, графического и неинтерактивного режимов). Кроме того, можно создать центральную базу данных ответов для распространения одной конфигурации на несколько компьютеров. Наиболее важным является то, что теперь возможно задать пользователю все вопросы подряд, до начала длительного процесса установки или обновления. Пользователь может отойти по своим делам, пока система занимается собственно установкой, а не глядеть неотрывно на экран в ожидании вопросов.

Контрольные суммы, конфигурационные файлы

Помимо сценариев разработчика и контрольных данных, уже рассмотренных выше, архив `control.tar.gz` пакета `Debian` может содержать другие интересные файлы:

```
# ar p /var/cache/apt/archives/bash_4.4-2_amd64.deb control.tar.gz | tar -tzf
-
./
./conffiles
./control
./md5sums
./postinst
./postrm
./preinst
./prerm
```

Первый, `md5sums`, содержит контрольные суммы MD5 для всех файлов пакета. Благодаря данному файлу можно с помощью команды `dpkg --verify` проверить, изменялись ли эти файлы с момента установки. Обратите внимание: при отсутствии этого файла `dpkg` создаст его динамически во время установки (и сохранит в базе данных `dpkg`, как и другие контрольные файлы).

В файле `conffiles` содержится список файлов, которые нужно обработать как конфигурационные. Таковые администратор может изменить, и `dpkg` постарается сохранить эти изменения во время обновления пакета.

Действительно, в этой ситуации `dpkg` ведет себя настолько разумно, насколько это возможно: если стандартный конфигурационный файл не изменился между двумя версиями, то программа ничего не делает. Однако в противном случае она будет пытаться обновить его. Возможны два варианта развития событий: если администратор не трогал конфигурационный файл, то `dpkg` автоматически установит новую версию; при наличии же изменений `dpkg` спросит администратора, какую версию тот хочет использовать (старую с изменениями или новую из пакета). Для помощи в принятии решения `dpkg` показывает `diff`, то есть различия между двумя версиями. Если пользователь предпочтет оставить старую версию, то новая будет храниться в том же месте, в файле с суффиксом `.dpkg-dist`. Если же пользователь выбирает новую версию, то старая сохраняется в файле с суффиксом `.dpkg-old`. Другой вариант заключается в том, чтобы немедленно прервать работу `dpkg` и отредактировать файл, попытавшись внести нужные изменения (ранее обнаруженные с помощью `diff`).

Хотя `dpkg` самостоятельно выполняет обновление конфигурационных файлов, программа все же регулярно прерывает работу, запрашивая ввод у администратора. Это весьма малоприятно для тех, кто хочет, чтобы обновление выполнялось неинтерактивно. Как следствие, у программы имеются параметры, позволяющие системе выбирать ответы автоматически, руководствуясь одной и той же логикой: `--force-confold` оставляет старую версию файла; `--force-confnew` использует более новую версию файла (данный выбор применяется, даже если файл не изменялся администратором, что крайне редко является желаемым эффектом). Добавление параметра `--force-confdef` указывает `dpkg`, что решения должны по возможности приниматься автоматически (в тех случаях, когда конфигурационный

файл не менялся), а `--force-confnew` или `--force-confold` надо использовать в остальных ситуациях.

Эти параметры применимы для `dpkg`, но администратор чаще имеет дело с программами `aptitude` или `apt-get`. Так что важно знать синтаксис, используемый для передачи параметров команде `dpkg` (их интерфейсы командной строки очень похожи).

```
# apt -o DPkg::options::="--force-confdef" -o DPkg::options::="--force-
    ➥confold" full-upgrade
```

Указанные параметры можно сохранить непосредственно в конфигурации `apt`. Для этого нужно добавить следующую строку в файл `/etc/apt/apt.conf.d/local`:

```
DPkg::options { "--force-confdef"; "--force-confold"; }
```

Включение данного параметра в файл конфигурации означает, что он также будет использоваться в графическом интерфейсе, в частности в `aptitude`.

И наоборот, вы можете заставить `dpkg` всегда задавать вопросы по поводу конфигурационных файлов. Параметр `--force-confask` вынуждает `dpkg` отображать вопросы о конфигурационных файлах даже в тех случаях, когда в этом обычно нет необходимости. Таким образом, при переустановке пакета с данным параметром `dpkg` будет задавать вопросы снова и снова для всех конфигурационных файлов, измененных администратором. Это очень удобно, особенно для переустановки оригинального конфигурационного файла, если он был удален и никакой другой экземпляр не доступен: обычная переустановка тут не сработает, так как `dpkg` считает удаление формой нормального изменения и поэтому не устанавливает желанный конфигурационный файл.

8.5. Резюме

В этой главе мы подробнее рассмотрели систему пакетов Debian, обсудили инструменты APT и `dpkg`, поговорили об основном взаимодействии пакетов, о дополнительной настройке и использовании APT и углубились в систему пакетов Debian с небольшой отсылкой к файлам формата `.deb`. Мы рассмотрели файл `control`, сценарии конфигурации, контрольные суммы и файл `conffiles`.

- Пакет Debian представляет собой сжатый архив программного приложения. Он содержит файлы приложения, а также другие метаданные, включая имена зависимостей, которые требуются приложениям, и сценарии, позволяющие выполнять команды на разных этапах жизненного цикла пакета (установка, удаление, обновление).
- Инструмент `dpkg`, в отличие от `apt` и `apt-get` (семейства APT), не знает всех доступных пакетов, которые могут использоваться для выполнения зависимостей пакетов. Таким образом, для управления пакетами Debian лучше применять последние утилиты, поскольку они могут автоматически разрешать проблемы с зависимостями.

- Можно использовать APT для установки и удаления приложений, обновления пакетов и даже обновления всей системы. Ниже указаны основные моменты, которые вы должны знать об APT и его конфигурации.
  - Файл `sources.list` является ключевым файлом конфигурации для определения источников пакетов (или репозиториев, содержащих пакеты).
  - Debian и Kali используют три раздела для дифференциации пакетов в соответствии с лицензиями, выбранными авторами каждого из проектов. Раздел `main` содержит все пакеты, полностью соответствующие критериям Debian по определению свободного ПО; `non-free` включает программное обеспечение, которое не (полностью) соответствует этим критериям, но тем не менее может быть распространено без ограничений; `contrib` (contributions) представляет собой набор программ с открытым исходным кодом, которые не способны функционировать без некоторых non-free-элементов.
  - Kali поддерживает несколько репозиториев, в том числе: kali-roll, который является основным хранилищем для конечных пользователей и всегда должен содержать самые новые и устанавливаемые пакеты; kali-dev, используемый разработчиками Kali и не предназначенный для публичного использования; и kali-bleeding-edge, который часто включает непроверенные и не протестированные пакеты, автоматически создаваемые из хранилища Git (или Subversion) менее чем через 24 часа после их публикации.
  - При работе с APT вы должны сначала скачать список доступных в настоящее время пакетов с помощью команды `apt update`.
  - Вы можете добавить пакет в систему с помощью команды `apt install` *пакет*. APT автоматически установит необходимые зависимости.
  - Чтобы удалить пакет, используйте `apt remove` *пакет*. Программа также устранит обратные зависимости пакета (то есть пакеты, зависящие от пакета, который нужно удалить).
  - Для удаления всех данных, связанных с пакетом, вы можете «очистить» пакет командой `apt purge` *пакет*. Такая «очистка» удалит не только пакет, но и его файлы конфигурации, а иногда и связанные с ним данные пользователя.

Мы рекомендуем проводить регулярные обновления для установки последних обновлений безопасности. Чтобы обновить систему, используйте `apt update`, а затем `apt upgrade`, `apt-get upgrade` или `aptitude safe-upgrade`. Эти команды ищут установленные пакеты, которые можно обновить без удаления каких-либо других пакетов.

Для более серьезных обновлений, таких как обновление основных версий, применяйте команду `apt full-upgrade`. С ее помощью `apt` завершит обновление, даже если ему придется удалить некоторые устаревшие пакеты или установить новые зависимости. Кроме того, данную команду вы должны использовать для регулярных обновлений вашей системы Kali Rolling. Ознакомьтесь со всеми преимуществами и недостатками обновлений, описанными в этой главе.

- Для проверки пакетов Debian можно использовать несколько инструментов:
  - `dpkg --listfiles` *пакет* (или `-L`) выводит на экран список файлов, которые были установлены указанным пакетом;
  - `dpkg --search` *файл* (или `-S`) находит все пакеты, которые содержат файл или путь, передаваемый в аргументе;
  - `dpkg --list` (или `-l`) отображает список известных системе пакетов и их статус установки;
  - `dpkg --contents` *файл.deb* (или `-c`) перечисляет все файлы, содержащиеся в заданном файле формата `.deb`;
  - `dpkg --info` *файл.deb* (или `-I`) выведет на экран заголовки указанного файла формата `.deb`;
  - различные подкоманды `apt-cache` отображают большую часть информации, хранящейся во внутренней базе данных APT.
- Чтобы избежать чрезмерного использования диска при частом обновлении, вы должны регулярно сортировать каталог `/var/cache/apt/archives/`. Для этого можно применять две команды: `apt clean` (или `apt-get clean`), полностью очищающую каталог; `apt autoclean` (или `apt-get autoclean`), удаляющую только те пакеты, которые больше нельзя скачать, поскольку они исчезли с зеркала и поэтому являются бесполезными.
- `Aptitude` — интерактивная программа, которая может использоваться в полуграфическом режиме на консоли. Это чрезвычайно надежная программа, которая поможет установить пакеты и устранить потенциальные неполадки.
- `synaptic` — графический менеджер пакетов, имеющий понятный и эффективный графический интерфейс.
- Как продвинутый пользователь вы можете создавать файлы по адресу `/etc/apt/apt.conf.d/` для настройки определенных аспектов APT. Вы также можете управлять приоритетами пакетов, отслеживать автоматически установленные пакеты, работать с несколькими дистрибутивами или архитектурами одновременно, задействовать криптографические подписи для проверки пакетов и обновлять файлы с помощью методов, описанных в этой главе.
- Несмотря на то что разработчики Kali/Debian усердно трудятся над тем, чтобы обновление системы проходило безболезненно, оно не всегда выполняется так гладко, как ожидается. При возникновении подобных ситуаций необходимо обратиться к системам отслеживания ошибок Kali и Debian по адресу https://bugs.debian.org/пакет и проверить, сообщал ли кто-нибудь ранее о проблеме. Кроме того, можно попытаться откатиться к предыдущей версии пакета или отладить и восстановить нерабочий сценарий поддержки пакета.

ГЛАВА

9

Расширенное использование системы



Ключевые темы:

- пользовательские пакеты;
- пользовательское ядро;
- пользовательские образы;
- пакет live-build;
- постоянное хранилище.

Kali создана как модульная настраиваемая среда, ориентированная на цели пентестинга, которая поддается глубокой настройке и поддерживает различные сценарии использования. Настраивать систему можно на различных уровнях, начиная с исходного кода. Исходный код пакетов Kali общедоступен. В данной главе мы покажем способы находить пакеты, модифицировать их и собирать из них собственные, настроенные так, как нужно именно вам. Модификация ядра Linux — своеобразная область настройки системы, поэтому ей посвящен отдельный раздел 9.2, в котором мы поговорим о том, где найти исходный код ядра, как настроить систему его сборки и, наконец, как его скомпилировать и собрать необходимые пакеты ядра.

Второй уровень настройки системы заключается в сборке live-образов. Мы покажем, как, используя инструмент `live-build`, задействовать большое количество возможностей по тонкой настройке готовых ISO-образов. В том числе речь пойдет о применении предварительно настроенных в соответствии с вашими нуждами пакетов Debian вместо пакетов, доступных на зеркалах.

Кроме того, мы поговорим о том, как создать постоянное хранилище данных для live-образа, записанного на USB-накопитель. Подобная конфигурация позволяет сохранять файлы и изменения операционной системы между перезагрузками.

9.1. Модификация пакетов Kali

Модификация пакетов Kali обычно является задачей для участников проекта и разработчиков: они обновляют пакеты до новых выпущенных версий, меняют стандартные конфигурации для лучшей интеграции пакетов в дистрибутив или исправляют указанные пользователями ошибки. Однако у вас могут быть особые потребности, которые официальные пакеты не способны удовлетворить, поэтому знание того, как собрать модифицированный пакет, может оказаться весьма полезным.

Возможно, вы зададитесь вопросом о том, почему вам вообще нужно возиться с пакетом. В конце концов, если требуется модифицировать какое-то программное обеспечение, то вы всегда можете загрузить его исходный код (обычно с помощью `git`), изменить и запустить измененную версию. Такая комбинация может сработать, но только когда она возможна и когда вы используете для данной цели личный каталог. Но если ваше приложение нуждается в установке, делающей его доступным во всей системе (например, с помощью команды `make install`), то оно засорит файловую систему файлами, неизвестными `dpkg`, и скоро станет источником проблем, которые невозможно обнаружить, основываясь на анализе зависимостей пакета. Более того, правильно подготовленные пакеты можно передавать кому-нибудь еще, их гораздо легче разворачивать на множестве компьютеров или отменять в них изменения после того, как обнаружилось, что они не работают, как ожидалось.

Итак, когда может понадобиться модификация пакетов? Рассмотрим несколько примеров.

Для начала представим, что вы интенсивно используете SET (Secure Electronic Transaction — безопасные электронные транзакции) и заметили, что разработчики

пакета выпустили новый релиз. Но все создатели Kali заняты участием в конференции, а вам срочно нужно его попробовать. В результате вам придется обновить пакет самостоятельно.

Или в другой ситуации допустим, что вам не удается заставить работать карту MIFARE NFC и вы хотите пересобрать `libfreefare` для того, чтобы включить отладочные сообщения и получить некие данные, которые можно включить в подготавливаемый вами отчет об ошибках. И наконец, представим, что программа `pyrit` прекращает работу с таинственным сообщением об ошибке. После поиска в Интернете вы обнаруживаете коммит в Git-репозитории создателей, который, как вам кажется, может исправить эту проблему, и вы хотите пересобрать пакет, включив в него найденное исправление.

Далее мы подробно рассмотрим эти ситуации и постараемся обобщить объяснения таким образом, чтобы вы могли воспользоваться представленными методиками и в других случаях, но все же невозможно рассказать обо всем, с чем у вас есть шанс встретиться. Если вы столкнулись с проблемой, то постарайтесь найти решение самостоятельно или обратитесь за помощью на подходящие форумы (см. главу 6).

Какими бы ни были изменения, которые вы хотите внести в пакет, общий процесс всегда один и тот же: загрузить исходный код пакета, распаковать его, внести изменения и собрать пакет. Но на каждом из этих шагов обычно можно воспользоваться множеством инструментов, предназначенных для решения указанных задач. Мы отобрали наиболее подходящие и наиболее популярные утилиты, однако данный список далеко не полный.

Загрузка исходного кода

Восстановление пакета Kali начинается с получения исходного кода. Исходный пакет состоит из нескольких файлов: основной файл имеет формат `*.dsc` (Debian Source Control), так как содержит список других сопровождающих файлов, которые могут быть формата `*.tar.gz`, `bz2`, `xz`, иногда `*.diff.gz` или `*.debian.tar.gz`, `bz2`, `xz`.

Исходные пакеты хранятся на зеркалах Kali, которые доступны через HTTP. Вы можете использовать браузер для скачивания всех необходимых файлов, но самый простой способ сделать это — применить команду `apt source` *`имя_исходного_пакета`*. Для нее требуется строка `deb-src` в файле `/etc/apt/sources.list` и актуальные индексные файлы (обновить их можно с помощью команды `apt update`). По умолчанию в Kali вышеописанные настройки не включены, так как очень немногим пользователям требуется исходный код пакетов, но привести все в нужное состояние очень просто. Для этого надо добавить строку `deb-src` в файл `/etc/apt/sources.list` (см. пример файла в подразделе «Репозитории Kali» и соответствующие пояснения в подразделе «Подробности о файле sources.list» раздела 8.1).

```
$ apt source libfreefare
Reading package lists... Done
NOTICE: 'libfreefare' packaging is maintained in the 'Git' version control
    ➥system at:
git://anonscm.debian.org/collab-maint/libnfc.git
```

```
Please use:
git clone git://anonscm.debian.org/collab-maint/libnfc.git
to retrieve the latest (possibly unreleased) updates to the package.
Need to get 119 kB of source archives.
Get:1 http://archive-2.kali.org/kali kali-rolling/main libfreefare 0.4.0-2 (dsc)
    ➥[2,090 B]
Get:2 http://archive-2.kali.org/kali kali-rolling/main libfreefare 0.4.0-2 (tar)
    ➥[113 kB]
Get:3 http://archive-2.kali.org/kali kali-rolling/main libfreefare 0.4.0-2
    ➥(diff) [3,640 B]
Fetched 119 kB in 1s (63.4 kB/s)
gpgv: keyblock resource '/home/rhertzog/.gnupg/trustedkeys.gpg': file open error
gpgv: Signature made Tue 04 Mar 2014 06:57:36 PM EST using RSA key ID 40AD1FA6
gpgv: Can't check signature: public key not found
dpkg-source: warning: failed to verify signature on ./libfreefare_0.4.0-2.dsc
dpkg-source: info: extracting libfreefare in libfreefare-0.4.0
dpkg-source: info: unpacking libfreefare_0.4.0.orig.tar.gz
dpkg-source: info: unpacking libfreefare_0.4.0-2.debian.tar.xz
$ cd libfreefare-0.4.0
$ ls
AUTHORS    CMakeLists.txt   COPYING HACKING            m4            README
ChangeLog  configure.ac     debian libfreefare         Makefile.am   test
cmake      contrib          examples libfreefare.pc.in NEWS          TODO
$ ls debian
changelog  copyright                libfreefare-dev.install  rules
compat     libfreefare0.install     libfreefare-doc.install  source
control    libfreefare-bin.install  README.Source            watch
```

В данном примере мы загружаем пакет с исходным кодом с зеркала Kali. Это такой же пакет, как и в Debian, поскольку строка версии не содержит подстроки `kali`. Это значит, что в данный пакет не было внесено каких-либо специфичных для Kali изменений.

Если вам нужна конкретная версия пакета с исходным кодом, которая на момент загрузки недоступна в репозиториях, перечисленных в `/etc/apt/sources.list`, то в таком случае легче всего скачать ее через URL ее `.dsc`-файла, найдя его на сайте http://pkg.kali.org/ и использовав его в команде `dget` (из пакета `devscripts`).

После нахождения URL для пакета с исходным кодом `libfreefare`, доступного в kali-bleeding-edge, вы можете скачать его с помощью `dget`. Сначала скачается файл `.dsc`, после чего он будет разобран для того, чтобы выяснить, на какие еще файлы он ссылается, затем скачаются и они:

```
$ dget http://http.kali.org/pool/main/libf/libfreefare/
    ➥libfreefare_0.4.0+0~git1439352548.ffde4d-1.dsc
dget: retrieving http://http.kali.org/pool/main/libf/libfreefare/
    ➥тlibfreefare_0.4.0+0~git1439352548.ffde4d-1.dsc
   % Total     % Received  % Xferd  Average Speed   Time    Time     Time
Current
                                 Dload  Upload   Total   Spent    Left   Speed
100   364  100   364    0    0    852      0 --:--:-- --:--:-- --:--:--   854
100  1935  100  1935    0    0   2650      0 --:--:-- --:--:-- --:--:-- 19948
```

```
dget: retrieving http://http.kali.org/pool/main/libf/libfreefare/
    ➥libfreefare_0.4.0+0~git1439352548.ffde4d.orig.tar.gz
[...]
dget: retrieving http://http.kali.org/pool/main/libf/libfreefare/
    ➥libfreefare_0.4.0+0~git1439352548.ffde4d-1.debian.tar.xz
[...]
libfreefare_0.4.0+0~git1439352548.ffde4d-1.dsc:
dscverify: libfreefare_0.4.0+0~git1439352548.ffde4d-1.dsc failed signature check:
gpg: Signature made Wed Aug 12 06:14:03 2015 CEST
gpg:                using RSA key 43EF73F4BD8096DA
gpg: Can't check signature: No public key
Validation FAILED!!
   $ dpkg-source -x libfreefare_0.4.0+0~git1439352548.ffde4d-1.dsc
gpgv: Signature made Wed Aug 12 06:14:03 2015 CEST
gpgv:                using RSA key 43EF73F4BD8096DA
gpgv: Can't check signature: No public key
dpkg-source: warning: failed to verify signature on ./
libfreefare_0.4.0+0~git1439352548.ffde4d-1.dsc
dpkg-source: info: extracting libfreefare in libfreefare-0.4.0+0~git1439352548.
    ➥ffde4d
dpkg-source: info: unpacking libfreefare_0.4.0+0~git1439352548.ffde4d.orig.tar.gz
dpkg-source: info: unpacking libfreefare_0.4.0+0~git1439352548.ffde4d-1.debian.
    ➥tar.xz
```

Стоит отметить, что `dget` не выполняет автоматическую распаковку пакетов с исходным кодом, так как не может проверить PGP-подписи этих пакетов. Таким образом, нужно выполнить это вручную, применив команду `dpkg-source -x dsc-file`. Кроме того, можно активировать принудительное извлечение пакетов с исходным кодом, задействовав параметр `--allow-unauthenticated` или `-u`. И наоборот, воспользоваться параметром `--download-only` для того, чтобы пропустить шаг распаковки.

Загрузка исходного кода из Git

Вероятно, вы заметили, что при вызове apt source вам сообщают о возможности использовать Git-репозиторий для поддержки пакета. Это может быть репозиторий Debian Git или Kali Git.

Все пакеты, специально подготовленные для Kali, можно обнаружить в Git-репозиториях, размещенных на git.kali.org. Загрузить код из этих репозиториев поможет команда git clone git://git.kali.org/packages/source-package. Если при выполнении данной команды не удается загрузить требуемое, то попробуйте переключиться на ветку kali/master, используя команду git checkout kali/master.

В отличие от того, что можно загрузить с помощью команды apt source, в полученном дереве не будут автоматически применены патчи. Взгляните на расположение debian/patches/ с целью узнать о возможных изменениях, сделанных в Kali.

```
$ git clone git://git.kali.org/packages/kali-meta
Cloning into 'kali-meta'...
```

```
remote: Counting objects: 760, done.
remote: Compressing objects: 100 % (614/614), done.
remote: Total 760 (delta 279), reused 0 (delta 0)
Receiving objects: 100 % (760/760), 141.01 KiB | 0 bytes/s, done.
Resolving deltas: 100 % (279/279), done.
Checking connectivity... done.
$ cd kali-meta
$ ls
debian
$ ls debian
changelog  compat  control  copyright  rules  source
```

Вы можете использовать git-репозитории как другой способ извлечения источников и, следовательно (в основном), соблюдать другие инструкции из данного раздела. Но в процессе взаимодействия с этими репозиториями создатели Kali применяют другой рабочий процесс пакетирования и инструменты из пакета git-buildpackage, о которых мы здесь не будем говорить. Вы можете узнать больше об этих инструментах по адресу https://honk.sigxcpu.org/piki/projects/git-buildpackage/.

Установка зависимостей для сборки

Итак, у вас есть исходный код, но нужно еще установить зависимости сборки. Они необходимы для того, чтобы собрать бинарный пакет, и, вероятно, пригодятся для частичных сборок, которые вам может понадобиться выполнять в целях проверки изменений при их внесении в пакет.

В каждом исходном пакете зависимости сборки объявлены в поле `Build-Depends` файла `debian/control`. Используем `apt` для установки этих зависимостей (предполагается, что вы находитесь в каталоге, содержащем распакованный исходный пакет):

```
$ sudo apt build-dep ./
Note, using directory './' to get the build dependencies
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
 autoconf automake autopoint autotools-dev debhelper dh-autoreconf
 dh-strip-nondeterminism gettext intltool-debian libarchive-zip-perl
 libfile-stripnondeterminism-perl libtool po-debconf
0 upgraded, 13 newly installed, 0 to remove and 0 not upgraded.
Need to get 4 456 kB of archives.
After this operation, 14,6 MB of additional disk space will be used.
Do you want to continue? [Y/n]
[…]
```

В этом примере все зависимости сборки можно разрешить с помощью пакетов, которые доступны APT. Так может быть не всегда, поскольку инструменты сборки

kali-rolling не проверяют вероятность установки зависимостей сборки (во внимание принимаются только зависимости бинарных пакетов). На практике бинарные зависимости и зависимости сборки часто тесно связаны, и для большинства пакетов достаточно вторых.

Внесение изменений

Весьма затруднительно рассказать обо всех вероятных видах изменений, которые может понадобиться внести в определенный пакет. Это потребовало бы освещения всех тонкостей работы с пакетами Debian (https://www.debian.org/doc/manuals/maint-guide/). Однако здесь мы расскажем о трех наиболее распространенных случаях, упомянутых выше, и опишем ряд важнейших шагов процесса модификации пакетов, которых не получится избежать (наподобие поддержки в актуальном состоянии файла `changelog`).

Первое, что надо сделать, — изменить номер версии пакета, чтобы новые пакеты можно было отличить от исходных, присутствующих в Kali или Debian. Для этого мы обычно добавляем суффикс, идентифицирующий исполнителя изменений (частное лицо или компанию). Так, например, мой ник в IRC — `buxy`, поэтому я применю его как суффикс. Подобное изменение лучше всего производить с помощью команды `dch` *(Debian CHangelog)* из пакета `devscripts`. В моем случае вызов данной команды будет выглядеть как `dch --local buxy`. Она вызывает текстовой редактор (`sensible-editor`, который запускает редактор, указанный в переменной окружения `VISUAL` или `EDITOR`, или в противном случае вызывается `/usr/bin/editor`), позволяющий задокументировать изменения, вносимые в конкретную сборку. В данном случае видно, что команда `dch` действительно изменила файл `debian/changelog`:

```
$ head -n 1 debian/changelog
libfreefare (0.4.0-2) unstable; urgency=low
$ dch --local buxy
[...]
$ head debian/changelog
libfreefare (0.4.0-2buxy1) UNRELEASED; urgency=medium

  * Enable --with-debug configure option.

 -- Raphael Hertzog <buxy@kali.org> Fri, 22 Apr 2016 10:36:00 -0400

libfreefare (0.4.0-2) unstable; urgency=low

  * Update debian/copyrtight.
    Fix license to LGPL3+.
```

Если вы производите такие изменения регулярно, то, возможно, есть смысл внести в переменные окружения `DEBFULLNAME` и `DEBEMAIL` ваше полное имя и адрес

электронной почты соответственно. Эти данные могут быть использованы множеством инструментов для работы с пакетами, включая dch, который внедрил их в строку, начинающуюся с символов -- в вышеприведенном примере.

Применение патча

В одном из наших случаев мы скачали исходный пакет pyrit и собираемся применить патч, который нашли в Git-репозитории разработчиков. Это довольно распространенная операция, и, как следствие, не должно возникнуть сложностей с ее выполнением. К сожалению, особенности работы с патчами могут различаться, что зависит от формата исходного пакета и от того, как именно организовано взаимодействие с пакетами в репозитории Git (в том случае, если для поддержки пакета используется Git).

Применение патча к нераспакованному исходному пакету. Итак, вы выполнили команду apt source pyrit, и у вас есть каталог pyrit-0.4.0. Можно применить патч напрямую, воспользовавшись командой patch -p1< *патч-файл*:

```
$ apt source pyrit
[...]
$ cd pyrit-0.4.0
$ wget https://github.com/JPaulMora/Pyrit/commit/
    ➥14ec997174b8e8fd20d22b6a97c57e19633f12a0.patch -O /tmp/pyrit-patch
[...]
$ patch -p1 </tmp/pyrit-patch
patching file cpyrit/pckttools.py
Hunk #1 succeeded at 53 (offset -1 lines).
$ dch --local buxy "Apply patch to work with scapy 2.3"
```

К этому моменту у вас имеется исходный код, пропатченный вручную, и вы можете собрать бинарный файл модифицированной версии пакета. Но если вы попытаетесь собрать обновленный пакет, то произойдет ошибка, в сообщении которой говорится о неожиданных изменениях кода пакета, подготовленного его разработчиками (unexpected upstream changes). Это произошло из-за того, что pyrit (как и большинство исходных пакетов) использует исходный формат (см. файл debian/source/format), известный как 3.0 (quilt). В нем изменения в коде должны быть записаны в отдельных патчах, хранимых в debian/patches/, притом файл debian/patches/series указывает на порядок, в котором эти патчи нужно применить. Вы можете зарегистрировать внесенные изменения в виде нового патча, воспользовавшись командой dpkg-source --commit:

```
$ dpkg-source --commit
dpkg-source: info: local changes detected, the modified files are:
 pyrit-0.4.0/cpyrit/pckttools.py
Enter the desired patch name: fix-for-scapy-2.3.patch
dpkg-source: info: local changes have been recorded in a new patch:
    ➥pyrit-0.4.0/debian/patches/fix-for-scapy-2.3.patch
$ tail -n 1 debian/patches/series
fix-for-scapy-2.3.patch
```

> **Управление патчами с помощью quilt**
>
> Соглашение по работе с патчами стало популярным благодаря инструменту quilt. Формат исходного пакета 3.0 (quilt), таким образом, совместим с данным инструментом — с небольшим изменением, которое заключается в том, что он использует debian/patches вместо patches. Этот инструмент можно найти в одноименном пакете, а перейдя по ссылке https://raphaelhertzog.com/2012/08/08/how-to-use-quilt-to-manage-patches-in-debian-packages/, ознакомиться с руководством.

Если пакет с исходным кодом использует формат 1.0 или 3.0 (native), то требования по регистрации изменений в коде пакетов в виде патчей отсутствуют. Сведения об изменениях автоматически встраиваются в получившийся исходный пакет.

Применение патча к коду, полученному из Git-репозитория. Если вы задействовали Git для загрузки исходного кода, то ситуация осложняется. Существует множество способов организации работы с Git и связанными с ними инструментами, и очевидно, что далеко не все пакеты Debian готовят с помощью одних и тех же рабочих процессов и инструментов. Рассмотренное выше различие пакетов, касающееся формата файлов, применимо и здесь, но, работая с Git, необходимо выяснить, задействованы ли уже патчи в дереве исходного кода или только хранятся в `debian/patches` (в этом случае они применяются в ходе сборки).

Самый популярный инструмент — *git-buildpackage*. Именно его мы применяем для управления репозиториями на git.kali.org. Когда вы используете его, патчи не применяются предварительно в дереве исходного кода, а вместо этого хранятся в `debian/patches`. Вы можете вручную добавить патчи в данный каталог и внести их список в `debian/patches/series`, но пользователи git-buildpackage обычно задействуют `gbp pq` для редактирования всей последовательности патчей как одной ветки, которую вы можете расширить или перекомпоновать в соответствии со своими потребностями. Посмотрите справку, выполнив команду `gbp-pq(1)`, чтобы разобраться, как это сделать.

Для работы с пакетами в Git имеется еще один инструмент — *git-dpm* (и команда с тем же именем). Он записывает метаданные в файл `debian/.git-dpm` и поддерживает в актуальном состоянии патчи в дереве исходного кода, используя команду `rebase` в применении к ветке, которую собирает из содержимого `debian/patches`.

Настройка параметров сборки

Обычно параметры сборки приходится настраивать в тех случаях, когда нужно включить дополнительные функции или особенности поведения пакета, которые не активированы в его официальном варианте. Это бывает необходимо и в ситуациях, требующих особого значения параметров, задаваемых во время сборки с помощью опции `./configure` или переменных, устанавливаемых в окружении сборки.

В подобных случаях изменения обычно ограничены `debian/rules`, где задается последовательность шагов процесса сборки пакета. В самых простых случаях

строки, относящиеся к исходной конфигурации (`./configure` ...), или сами команды сборки (`$(MAKE)` ... либо `make` ...) найти несложно. Если эти команды не вызываются непосредственно, то, вероятно, их вызов выполняется через другие команды, вызываемые явно. В подобных случаях стоит обратиться к их документации, чтобы узнать о способах изменения стандартного поведения команд. При работе с пакетами, которые используют `dh`, вам может понадобиться переопределить команды `dh_auto_configure` или `dh_auto_build` (обратитесь к соответствующим руководствам за информацией о том, как это сделать).

Чтобы приблизить эти объяснения к практике, применим их к одному из вышеупомянутых случаев. А именно: мы собираемся модифицировать `libfreefare`, передав параметр `--enable-debug` сценарию `./configure` с целью получить в результате больше сведений от инструментов для работы с NFC и подготовить более качественный отчет по карте MIFARE NFC, которую не удается распознать. Поскольку пакет для управления процессом сборки использует `dh`, то нужно добавить (или, в данном случае, модифицировать) цель `override_dh_auto_configure`. Ниже представлено соответствующее извлечение из `libfreefare`-файла `debian/rules`:

```
override_dh_auto_configure:
      dh_auto_configure -- --without-cutter --disable-silent-rules --enable-debug
```

Пакетирование новой официальной версии

Взглянем на еще один пример, раз уж говорим о пакетировании официальных версий пакетов. Предположим, вы — опытный пользователь SET и заметили, что вышел новый официальный релиз (7.4.5), который пока недоступен в Kali (тут есть лишь версия 7.4.4). Вам хочется собрать и опробовать обновленный официальный пакет. Так как произошло лишь небольшое увеличение версии пакета, то вы не ожидаете, что изменение потребует каких-либо новшеств на уровне пакетирования.

Для обновления кода следует извлечь новый архив рядом с текущим исходным пакетом и скопировать каталог `debian` из текущего пакета в новый. Затем нужно поднять версию в `debian/changelog`.

```
$ apt source set
Reading package lists... Done
NOTICE: 'set' packaging is maintained in the 'Git' version control system at:
git://git.kali.org/packages/set.git
Please use:
git clone git://git.kali.org/packages/set.git
to retrieve the latest (possibly unreleased) updates to the package.
Need to get 42.3 MB of source archives.
[...]
dpkg-source: warning: failed to verify signature on ./set_7.4.4-0kali1.dsc
dpkg-source: info: extracting set in set-7.4.4
dpkg-source: info: unpacking set_7.4.4.orig.tar.gz
dpkg-source: info: unpacking set_7.4.4-0kali1.debian.tar.xz
```

```
dpkg-source: info: applying edit-config-file
dpkg-source: info: applying fix-path-interpreter.patch
$ wget https://github.com/trustedsec/social-engineer-toolkit/archive/7.4.5.tar.
    ➥gz –O set_7.4.5.orig.tar.gz
[...]
$ tar xvf set_7.4.5.orig.tar.gz
[...]
social-engineer-toolkit-7.4.5/src/wireless/wifiattack.py
$ cp -a set-7.4.4/debian social-engineer-toolkit-7.4.5/debian
$ cd social-engineer-toolkit-7.4.5
$ dch -v 7.4.5-0buxy1 "New upstream release"
```

Вот и все. Теперь можно собрать обновленный пакет.

В зависимости от изменений, внесенных в новую сборку официальной версии, может понадобиться вдобавок изменить зависимости сборки и времени выполнения, установить новые файлы. Это более сложные операции, которые здесь не рассматриваются.

Запуск сборки

Когда в исходный код внесены все необходимые изменения, вы можете приступить к созданию бинарных файлов в формате `.deb`. Весь этот процесс проходит под управлением команды `dpkg-buildpackage` и выглядит следующим образом:

```
$ dpkg-buildpackage -us -uc -b
dpkg-buildpackage: source package libfreefare
dpkg-buildpackage: source version 0.4.0-2buxy1
dpkg-buildpackage: source distribution UNRELEASED
dpkg-buildpackage: source changed by Raphael Hertzog <buxy@kali.org>
dpkg-buildpackage: host architecture amd64
[...]
  dh_builddeb
dpkg-deb: building package 'libfreefare0-dbgsym' in '../libfreefare0-
dbgsym_0.4.0-2buxy1_amd64.deb'.
dpkg-deb: building package 'libfreefare0' in '../libfreefare0_0.4.0-2buxy1_
amd64.deb'.
dpkg-deb: building package 'libfreefare-dev' in '../libfreefare-dev_0.4.0-
2buxy1_amd64.deb'.
dpkg-deb: building package 'libfreefare-bin-dbgsym' in '../libfreefare-bin-
dbgsym_0.4.0-2buxy1_amd64.deb'.
dpkg-deb: building package 'libfreefare-bin' in '../libfreefare-bin_0.4.0-
2buxy1_amd64.deb'.
dpkg-deb: building package 'libfreefare-doc' in '../libfreefare-doc_0.4.0-
2buxy1_all.deb'.
 dpkg-genchanges -b >../libfreefare_0.4.0-2buxy1_amd64.changes
dpkg-genchanges: binary-only upload (no source code included)
 dpkg-source --after-build libfreefare-0.4.0
dpkg-buildpackage: binary-only upload (no source included)
```

Параметры `-us -uc` отключают подписи для отдельных сгенерированных файлов (`.dsc`, `.changes`), так как эта операция даст сбой, если у вас нет ключей GnuPG, связанных с идентификационными данными, помещенными в файл `changelog`. Параметр `-b` предназначен для организации процесса сборки, на выходе которого оказываются только бинарные файлы. При подобном подходе создается не исходный пакет (`.dsc`), а только бинарные (`.deb`) пакеты. Используйте данный параметр, чтобы избежать сбоев на этапе сборки исходных пакетов: если вы не зарегистрировали соответствующим образом изменения в системе управления патчами, то в ходе этой операции могут возникнуть предупреждения и процесс сборки будет прерван.

Как можно узнать из сообщений `dpkg-deb`, созданные бинарные пакеты теперь доступны в родительском каталоге (в том, в котором находится каталог пакета с исходным кодом). Устанавливают такие пакеты с помощью команды `dpkg -i` или `apt install`.

```
$ sudo apt install ../libfreefare0_0.4.0-2buxy1_amd64.deb \
  ../libfreefare-bin_0.4.0-2buxy1_amd64.deb
Reading package lists... Done
Building dependency tree
Reading state information... Done
Note, selecting 'libfreefare0' instead of '../libfreefare0_0.4.0-2buxy1_amd64.deb'
Note, selecting 'libfreefare-bin' instead of '../libfreefare-bin_0.4.0-2buxy1_
amd64.deb'
The following packages will be upgraded:
  libfreefare-bin libfreefare0
2 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B/69,4 kB of archives.
After this operation, 2 048 B of additional disk space will be used.
[...]
```

Мы рекомендуем применять `apt install`, а не `dpkg -i`, поскольку эта команда позволяет легко справляться с проблемой отсутствующих зависимостей. Правда, не так давно приходилось использовать `dpkg` ввиду того, что команда `apt` не могла работать с файлами формата `.deb`, которые расположены вне репозиториев.

Программы-оболочки для dpkg-buildpackage

Чаще всего разработчики Debian для сборки пакетов используют программы-оболочки наподобие debuild. Эта программа запускает dpkg-buildpackage, как обычно, но, помимо этого, вызывает программу (lintian), которая выполняет множество проверок сгенерированного пакета на соответствие политике Debian. Данный сценарий также очищает окружение, в результате локальные переменные окружения не могут повлиять на процесс сборки пакета. Команда debuild — один из инструментов, входящих в состав пакета devscripts, элементы которого облегчают жизнь тем, кто занимается разработкой и поддержкой пакетов.

9.2. Перекомпиляция ядра Linux

Стандартные ядра Kali включают максимум функций и все доступные наборы драйверов. Так сделано для того, чтобы система могла работать на как можно большем количестве существующих конфигураций аппаратного обеспечения. Именно поэтому некоторые пользователи предпочитают перекомпилировать ядро с целью оставить в нем только то, что нужно именно им. Подобное решение обусловлено двумя причинами. Первая — оптимизация потребления памяти, поскольку весь код ядра, даже если не используется, занимает физическую память. Так как статически скомпилированные фрагменты ядра никогда не перемещаются в область подкачки, общее снижение системной производительности будет происходить из-за наличия встроенных драйверов и функций, которые никогда не применяются. Вторая причина заключается в том, что уменьшение числа драйверов и механизмов ядра снижает риск возникновения проблем с безопасностью, поскольку используется лишь часть доступного кода ядра.

Важно!



Если вы решили собрать собственное ядро, то должны понимать, что ответственность за последствия ложится на вас. Команда Kali не сможет обеспечить обновления безопасности для вашего ядра. Работая с ядром, предоставленным Kali, вы пользуетесь обновлениями, подготовленными в рамках Debian Project.

Кроме того, перекомпиляция ядра необходима в случае, если вам нужно воспользоваться определенными возможностями, которые доступны только в виде патчей (и не включены в стандартную версию ядра).

Руководство по ядру Debian

Команда по работе с ядром Debian поддерживает в актуальном состоянии руководство Debian Linux Kernel Handbook (оно также доступно в виде пакета debian-kernel-handbook) (https://kernel-team.pages.debian.net/kernel-handbook/). Это подробная документация, посвященная описанию большинства задач, связанных с ядром, и тому, как поддерживаются официальные пакеты ядра Debian. Если вам нужны подробности о сборке собственного ядра, то с Debian Linux Kernel Handbook стоит ознакомиться в первую очередь.

Подготовка и предварительные требования

Debian и Kali поддерживают ядро в форме пакета, что неудивительно, но такой метод отличается от традиционного подхода к компиляции и установке ядра. Поскольку ядро находится под контролем системы управления пакетами, то его можно без проблем удалить или развернуть на нескольких машинах. Более того, сценарии,

связанные с этими пакетами, автоматизируют взаимодействие с загрузчиком операционной системы и генератором initrd.

Официальный исходный код Linux содержит все необходимое для сборки пакета ядра Debian, но сначала нужно установить пакет `build-essential`, чтобы обеспечить наличие инструментов для сборки пакетов Debian. Более того, настройка ядра требует наличия пакета `libncurses5-dev`. И наконец, пакет `fakeroot` позволяет создавать пакеты Debian, не имея административных привилегий.

Для установки вышеупомянутых пакетов воспользуйтесь следующей командой:

```
# apt install build-essential libncurses5-dev fakeroot
```

Загрузка исходного кода

Поскольку исходные коды ядра Linux доступны в виде пакета, то вы можете их загрузить, установив пакет `linux-source-version`. Команда `apt-cache search ^linux-source` позволяет вывести список последних версий ядра Kali. Обратите внимание: исходный код, содержащийся в этих пакетах, отличается от того, что публикует Линус Торвальдс и разработчики ядра (https://www.kernel.org/). Как и все дистрибутивы, Debian и Kali применяют ряд патчей, которые могут и присутствовать в официальной версии Linux, и не присутствовать. Эти модификации включают бэкпорты исправлений, функций и драйверов из более новых версий ядра, новые функции, еще не полностью интегрированные в официальный код Linux, а иногда даже изменения, специфичные для Debian и Kali.

Ниже мы рассмотрим работу с ядром Linux версии 4.9, но наши примеры, конечно, можно адаптировать к той версии ядра, которая требуется вам.

Приводя этот пример, мы предполагаем, что установлен бинарный пакет `linux-source-4.9`. Обратите внимание: мы устанавливаем бинарный пакет, содержащий официальный исходный код, но не загружаем пакет с исходным кодом Kali, который называется `linux`.

```
# apt install linux-source-4.9
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  bc libreadline7
Suggested packages:
  libncurses-dev | ncurses-dev libqt4-dev
The following NEW packages will be installed:
  bc libreadline7 linux-source-4.9
0 upgraded, 3 newly installed, 0 to remove and 0 not upgraded.
Need to get 95.4 MB of archives.
After this operation, 95.8 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
[...]
# ls /usr/src
linux-config-4.9 linux-patch-4.9-rt.patch.xz linux-source-4.9.tar.xz
```

Обратите внимание: пакет содержит /usr/src/linux-source-4.9.tar.xz, сжатый архив источников ядра. Вы должны извлечь эти файлы в новый каталог (не прямо под /usr/src/, поскольку нет необходимости в специальных разрешениях для компиляции ядра Linux). Вместо этого используйте каталог ~/ kernel/ как более подходящий.

```
$ mkdir ~/kernel; cd ~/kernel
$ tar -xaf /usr/src/linux-source-4.9.tar.xz
```

Настройка ядра

Следующий шаг заключается в настройке ядра в соответствии с вашими нуждами. Точная процедура зависит от целей того, кто занимается сборкой нового ядра.

Процесс сборки зависит от конфигурационного файла ядра. В большинстве случаев имеет смысл как можно меньше отклоняться от стандартного конфигурационного файла Kali, который, как и во всех дистрибутивах Linux, устанавливается в каталог /boot. В этом случае, вместо того чтобы перенастраивать все с нуля, достаточно сделать копию файла /boot/config-*версия*. (Версия должна быть точно такой же, как версия используемого в данный момент ядра, которую можно выяснить с помощью команды uname -r.) Поместите копию в файл .config, расположенный в каталоге, содержащем исходный код ядра.

```
$ cp /boot/config-4.9.0-kali1-amd64 ~/kernel/linux-source-4.9/.config
```

В качестве альтернативы, ввиду того что имеется стандартная конфигурация ядра в arch/*архив*/configs/*_defconfig, можно поместить выбранную конфигурацию в требуемое место с помощью такой команды, как make x86_64_defconfig (в случае с 64-разрядным компьютером) или make i386_defconfig (для 32-разрядного компьютера).

Если вам не нужны изменения конфигурации, то здесь можно остановиться и перейти к подразделу «Компиляция и сборка пакета» (см. ниже). Если же вам требуется внести изменения или вы решили перенастроить все с нуля, то придется заняться настройками. В каталоге с исходным кодом существуют специальные средства для настройки ядра. Для их использования понадобится команда вида make *цель*, где *цель* — это название одного из инструментов, описанных ниже.

Команда make menuconfig компилирует и запускает текстовой интерфейс, предназначенный для конфигурирования ядра (именно здесь требуется пакет libncurses5-dev). Этот интерфейс дает доступ ко множеству настроек ядра, представленных в виде иерархической структуры. Нажатие клавиши Пробел позволяет изменить значение выбранной опции. Клавишей Enter «нажимают» кнопки, которые выбирают в нижней части экрана. Кнопка Select (Выбрать) в нижней части экрана применяется для перехода в выбранное подменю. Кнопка Exit (Выход) закрывает текущий экран и выполняет переход вверх по иерархии. Кнопка Help (Справка) выводит более подробные сведения о выбранной опции. Клавиши-стрелки позволяют перемещаться по списку опций и экранных кнопок.

Чтобы выйти из конфигурационной программы, выберите команду Exit (Выход) в главном меню. Затем программа предложит сохранить внесенные изменения, сделайте это, если вас все устраивает.

Другие средства имеют похожие возможности, но оформлены они в виде графических приложений. Например, команда `make xconfig` использует графический интерфейс, основанный на Qt, команда `make gconfig` задействует GTK+. Первая из этих двух команд требует наличия `libqt4-dev`, в то время как вторая зависит от `libglade2-dev` и `libgtk2.0-dev`.

Работа с устаревшими файлами .config

При использовании файла .config, сгенерированного для другой (обычно более старой) версии ядра, необходимо обновить его. Сделать это можно с помощью команды make oldconfig, которая в интерактивном режиме задаст ряд вопросов о новых параметрах конфигурации. Если вы хотите применить ответы по умолчанию на все вопросы, то можете задействовать команду make olddefconfig. Она позволяет автоматически ответить отрицательно на все вопросы.

Компиляция и сборка пакета

Очистка перед пересборкой

Если вы уже компилировали ранее ядро в каталоге, с которым работаете, и хотите перестроить все с нуля (например, потому что значительно изменили конфигурацию ядра), то вам нужно выполнить команду make clean для удаления скомпилированных файлов. Команда make distclean удаляет еще больше сгенерированных файлов, включая файл .config. Поэтому перед операцией очистки на всякий случай сделайте резервную копию данного файла.

После того как настройка ядра завершена, нам понадобится простая команда `make deb-pkg`. Она позволяет сгенерировать до пяти пакетов Debian в стандартном формате `.deb`: `linux-image-`*версия* содержит образ ядра и соответствующие модули; `linux-headers-`*версия* включает заголовочные файлы, требуемые для сборки внешних модулей; в `linux-firmware-image-`*версия* находятся файлы прошивок, которые нужны тем или иным драйверам (этого пакета может и не быть, если вы собираете ядро из исходников, взятых из Debian или Kali); `linux-image-`*версия*`-dbg` содержит отладочные символы для образа ядра и его модулей; `linux-libc-dev` включает заголовки, относящиеся к отдельным библиотекам пространства пользователя, вроде GNU C (glibc).

Значение `версия` в именах файлов задается в виде комбинации официальной версии (как задано в переменных `VERSION`, `PATCHLEVEL`, `SUBLEVEL` и `EXTRAVERSION` в `Makefile`), конфигурационного параметра `LOCALVERSION` и переменной окружения

LOCALVERSION. При формировании версии пакета используется та же строка версии с присоединенным к ней номером ревизии, который регулярно увеличивается (и хранится в .version), если только вы не переопределили этот номер с помощью переменной среды KDEB_PKGVERSION.

```
$ make deb-pkg LOCALVERSION=-custom KDEB_PKGVERSION=$(make kernelversion)-1
[...]
$ ls ../*.deb
../linux-headers-4.9.0-kali1-custom_4.9.2-1_amd64.deb
../linux-image-4.9.0-kali1-custom_4.9.2-1_amd64.deb
../linux-image-4.9.0-kali1-custom-dbg_4.9.2-1_amd64.deb
../linux-libc-dev_4.9.2-1_amd64.deb
```

Чтобы можно было воспользоваться собранным ядром, остался лишь один шаг — установить требуемые пакеты с помощью команды dpkg -i *файл*.deb. Здесь необходим пакет "linux-image". При наличии внешних модулей ядра для сборки понадобится установить пакет "linux-headers". Так бывает, если установлены некоторые пакеты "*-dkms" (проверить это позволяет команда dpkg -l "*-dkms" | grep ^ii). Другие пакеты в большинстве случаев не нужны.

9.3. Сборка собственных ISO-образов Kali

Kali Linux отличается гибкостью и имеет большое количество функциональных свойств. Сразу после его установки можно приступать к решению множества задач при наличии некоторого уровня знания инструментов, креативности, терпения и опыта. Однако образ Kali можно настраивать, включая в него то, что вам нужно, или убирая лишнее, задавать автоматическое выполнение каких-либо действий в ходе загрузки системы.

Примерами таких образов являются Kali ISO of Doom (https://www.offensive-security.com/kali-linux/kali-linux-iso-of-doom/) и Kali Evil Wireless Access Point (https://www.offensive-security.com/kali-linux/kali-linux-evil-wireless-access-point/). Это замечательные проекты, которые полагаются на специально настроенные реализации дистрибутива. Взглянем на процесс создания собственных ISO-образов Kali Linux.

Официальные образы Kali собраны с помощью live-build. Это набор сценариев, позволяющий полностью автоматизировать и настроить все аспекты создания ISO-образов. Набор live-build использует при формировании своей конфигурации структуру каталогов. Эту конфигурацию и некоторые связанные вспомогательные сценарии можно найти в Git-репозитории live-build-config. Мы задействуем данный репозиторий как основу для построения образов, настроенных в соответствии с особыми требованиями.

Прежде чем продолжать, важно уяснить, что команды, показанные в этом разделе, предназначены для выполнения в актуальной версии Kali Linux. Если попытаться воспользоваться ими в системе, отличной от Kali, или в устаревшей версии Kali, то они, вероятнее всего, не будут работать исправно.

Предварительные требования к установке

Первый шаг заключается в установке необходимых пакетов и в загрузке Git-репозитория с конфигурацией Kali live-build:

```
# apt install curl git live-build
[...]
# git clone git://git.kali.org/live-build-config.git
[...]
# cd live-build-config
# ls
auto build_all.sh build.sh kali-config README
```

После этого вы уже можете создавать обновленные (но немодифицированные) ISO-образы Kali, используя команду `./build.sh --verbose`. Сборка займет немало времени, так как в ее ходе будут скачиваться все пакеты, которые необходимо включить в образ. После завершения данного процесса вы сможете найти новый ISO-образ в каталоге `images`.

Сборка live-образа с различными окружениями рабочего стола

Упаковщик `build.sh` из набора `live-build` отвечает за подготовку каталога `config`, который необходим `live-build`. Он помогает задавать различные конфигурации в зависимости от его параметра `--variant`.

Упаковщик создает каталог `config`, комбинируя файлы из `kali-config/common` и `kali-config/variant-X`, где *X* — это название варианта, заданного с помощью параметра `--variant`. Когда данный параметр не задан явно, в качестве названия варианта используется `default`.

Каталог `kali-config` содержит каталоги для наиболее популярных окружений рабочего стола:

- `e17` для Enlightenment;
- `gnome` для GNOME;
- `i3wm` для фреймового оконного менеджера i3;
- `kde` для KDE;
- `lxde` для LXDE;
- `mate` для Mate Desktop Environment;
- `xfce` для XFCE.

Вариант `light` несколько особый. Он основан на XFCE и используется для создания официального «облегченного» ISO-образа, который содержит сокращенный набор приложений.

Вы можете легко создать live-образ Kali, применяя в качестве окружения рабочего стола KDE:

```
# ./build.sh --variant kde --verbose
```

Вышеописанная концепция *вариантов* позволяет выполнять общую настройку системы, пользуясь наборами стандартных предустановок. Однако если вы найдете время почитать руководство по установке Debian (Debian Live System Manual), то в нем вы обнаружите множество других способов настройки образов, которые заключаются в изменении содержимого соответствующих подкаталогов в `kali-config`. Ниже мы рассмотрим несколько примеров.

Изменение набора установленных пакетов

После того как `live-build` запущен, он устанавливает все пакеты, перечисленные в файлах `package-lists/*.list.chroot`. Стандартная конфигурация, которую мы предоставляем, включает файл `package-lists/kali.list.chroot`, содержащий запись о пакете `kali-linux-full` (это основной метапакет, который включает в образ все пакеты Kali). Строку с упоминанием данного пакета можно закомментировать и использовать другой метапакет или составить собственный список из других пакетов. Кроме того, можно скомбинировать оба подхода, начав с метапакета и добавляя дополнительные необходимые пакеты.

Используя `package-lists`, вы можете добавлять в образ только те пакеты, которые уже доступны в официальном репозитории Kali. Однако при наличии собственных пакетов включить их в live-образ можно, поместив соответствующие файлы формата `.deb` в каталог `packages.chroot` (например, в `kali-config/config-gnome/packages.chroot`, если вы собираете вариант GNOME).

Метапакеты — это пустые пакеты, единственная задача которых — включать множество зависимостей от других пакетов. Как результат, они упрощают установку наборов пакетов, которые обычно устанавливают вместе. Исходный пакет kali-meta отвечает за сборку всех метапакетов, предоставляемых Kali Linux:

- `kali-linux` — базовая система (используется во всех остальных метапакетах);
- `kali-linux-full` — стандартная установка Kali Linux;
- `kali-linux-all` — метапакет, объединяющий все остальные метапакеты, равно как и другие пакеты (содержит практически все имеющееся в Kali, так что это просто огромный пакет!);
- `kali-linux-sdr` — инструменты для программно-определяемого радио (Software Defined Radio, SDR);
- `kali-linux-gpu` — средства, использующие видеокарту (GPU) для выполнения тяжелых вычислений;
- `kali-linux-wireless` — утилиты для исследования и анализа беспроводных сетей;
- `kali-linux-web` — программы для исследования веб-приложений;
- `kali-linux-forensic` — средства цифровой криминалистики (их используют для поиска улик при расследовании различных инцидентов);
- `kali-linux-voip` — инструменты VoIP (Voice Over IP);
- `kali-linux-pwtools` — утилиты для взлома паролей;
- `kali-linux-top10` — десять самых популярных инструментов;
- `kali-linux-rfid` — средства для работы с RFID.

Эти метапакеты позволяют создать собственный список пакетов для `live-build`. Полный список доступных метапакетов и программ, которые они включают, можно найти по адресу https://tools.kali.org/kali-metapackages.

> **Автоматизация настройки установленных пакетов** Для автоматизации настройки установленных пакетов вы можете предоставить файлы предварительных ответов Debconf (см. подраздел «Создание файла пресидинга» раздела 4.3) как файлы вида preseed/*.cfg. Они будут применены для настройки пакетов, установленных в файловой live-системе.

Использование хуков для настройки содержимого образа

Инструмент `live-build` предлагает хуки (Hooks), выполняемые на различных этапах процесса сборки. Хуки chroot — это сценарии, которые устанавливаются как файлы `hooks/live/*.chroot` в дереве конфигурации и выполняются с помощью `chroot`. В то время как `chroot` — команда, которая позволяет временно изменить корневой каталог операционной системы на выбранный каталог, она также используется для назначения каталога, содержащего полное (альтернативное) дерево файловой системы. В случае с `live-build` каталог `chroot` — место, в котором готовится файловая live-система. Так как приложения, запущенные с помощью `chroot`, не имеют доступа за пределы выбранного каталога, то же самое справедливо и для chroot-хуков: применять и модифицировать можно лишь то, что доступно в окружении chroot. Мы полагаемся на эти хуки для выполнения множества настроек, специфичных для Kali (см. `kali-config/common/hooks/live/kali-hacks.chroot`).

Бинарные хуки (`hooks/live/*.binary`) исполняются в контексте процесса сборки, в конце данного процесса, но не вызываются в ходе сборки с помощью `chroot`. Они позволяют модифицировать содержимое сборки ISO-образа, но не файловую live-систему, поскольку к этому моменту она уже создана. Эта возможность используется в Kali для выполнения некоторых изменений в стандартной конфигурации isolinux, созданной live-build. Например, взгляните на `config/common/hooks/live/persistence.binary`, где мы добавляем пункты загрузочного меню, предназначенные для включения постоянного хранилища данных.

Добавление файлов в ISO-образ или в файловую live-систему

Еще один весьма распространенный способ настройки образов заключается в добавлении файлов либо в файловую live-систему, либо в ISO-образ.

Добавлять файлы в файловую систему можно, помещая их туда, где они должны быть — в каталоге конфигурации `includes.chroot`. Например, есть стандарт-

ный файл `kali-config/common/includes.chroot/usr/lib/live/config/0031-root-password`, который в итоге оказывается расположенным в файловой live-системе по адресу `/usr/lib/live/config/0031-root-password`.

> **Хуки live-boot** Сценарии, установленные в /lib/live/config/XXXX-имя, выполняются сценарием инициализации пакета live-boot. Они перенастраивают многие аспекты системы так, чтобы те подходили для работы в live-режиме. Сюда вы можете добавить собственные сценарии для настройки своей live-системы во время работы. В частности, их используют, например, для реализации собственных параметров загрузки.

Добавлять файлы в ISO-образ можно, размещая их в каталоге конфигурации `includes.binary`, в тех местах, где они должны быть. Например, есть стандартный файл `kali-config/common/includes.binary/isolinux/splash.png`, который переопределяет фоновое изображение, используемое загрузчиком isolinux (оно хранится в файле `/isolinux/splash.png` в файловой системе ISO-образа).

9.4. Добавление постоянного хранилища в live-образ Kali в формате ISO с помощью USB-накопителя

Особенности постоянного хранилища

В текущем разделе мы рассмотрим шаги, необходимые для добавления постоянного хранилища информации на USB-накопитель с записанным на него Kali. Сущность файловой live-системы состоит в ее эфемерности. Все данные, сохраненные при работе с такой системой, исчезают после перезагрузки, как и настройки системы. Чтобы избежать подобной ситуации, можно использовать функцию *`live-boot`*, называемую постоянным хранилищем информации (persistence). Эта функция активируется в том случае, если параметры загрузки содержат ключевое слово `persistence`.

Так как модификация загрузочного меню является непростой задачей, Kali по умолчанию имеет два пункта меню, позволяющих включить постоянное хранилище: Live USB Persistence (Live-USB-хранилище) и Live USB Encrypted Persistence (Зашифрованное live USB-хранилище), как показано на рис. 9.1.

Когда данная функция активна, *`live-boot`* просканирует все разделы в поисках файловых систем, помеченных как `persistence` (это можно изменить с помощью параметра загрузки `persistence-label=`*`значение`*), и установщик создаст хранилище для каталогов, перечисленных в файле `persistence.conf`, обнаруженном в этом разделе (каждый каталог указывается в отдельной строке). Особое значение `/ union` позволяет включить полное сохранение всех каталогов с помощью каскадно-объединенного монтирования (union mount). При таком подходе создается дополнительный

уровень файловой системы, в котором сохраняются лишь изменения, вносимые в данные базовой файловой системы. Данные каталогов, не теряющиеся после перезагрузки, хранятся в файловой системе, которая содержит соответствующий файл `persistence.conf`.

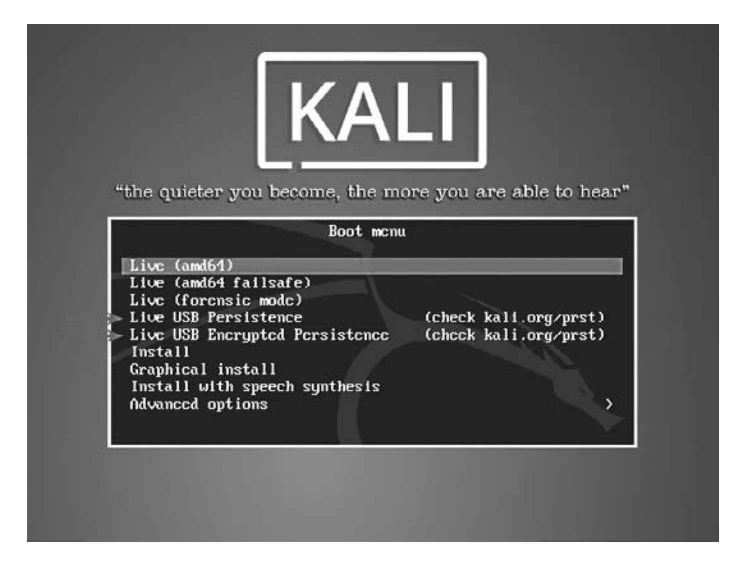


Рис. 9.1. Пункты меню для активации постоянного хранилища

Создание незашифрованного хранилища на USB-накопителе

Мы предполагаем, что вы подготовили USB-накопитель с live-системой, следуя инструкциям, которые можно найти в подразделе «Копирование образа на DVD- или USB-накопитель» раздела 2.1, и что размера носителя достаточно для хранения образа (около 3 Гбайт) и каталогов, которые попадут в постоянное хранилище. Мы также рассчитываем на то, что Linux распознает USB-накопитель как `/dev/sdb` и этот носитель содержит лишь два раздела, являющихся частью стандартного ISO-образа (`/dev/sdb1` и `/dev/sdb2`). Будьте осторожны при выполнении этой процедуры. Если вы случайно повторно разделите не тот диск, то можете потерять важные данные.

Чтобы добавить на диск новый раздел, необходимо знать размер скопированного образа; это позволит начать новый раздел сразу после live-образа. Затем нужно воспользоваться командой `parted` для создания раздела. Команды, приведенные ниже, выполняют анализ ISO-образа с именем `kali-linux-2016.1-amd64.iso`, присутствие которого ожидается на USB-накопителе:

```
# parted /dev/sdb print
Model: SanDisk Cruzer Edge (scsi)
Disk /dev/sdb: 32,0GB
```

```
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start   End                             Size    Type    File system
Flags
 1      32,8kB  2852MB  2852MB  primary               boot, hidden
 2      2852MB   2945MB 93,4MB  primary
# start=$(du --block-size=1MB kali-linux-2016.1-amd64.iso | awk '{print $1}')
# echo "Size of image is $start MB"
Size of image is 2946 MB
# parted -a optimal /dev/sdb mkpart primary "${start}MB" 100 %
Information: You may need to update /etc/fstab.

# parted /dev/sdb print
Model: SanDisk Cruzer Edge (scsi)
Disk /dev/sdb: 32,0GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start   End     Size    Type    File system   Flags
 1      32,8kB  2852MB  2852MB  primary               boot, hidden
 2      2852MB   2945MB 93,4MB  primary
 3      2946MB   32,0GB 29,1GB  primary
```

Когда новый раздел `/dev/sdb3` создан, отформатируйте его в файловой системе ext4 и назначьте ему метку `persistence` с помощью команды `mkfs.ext4` (и параметра `-L` для назначения метки). Затем монтируйте раздел в каталог `/mnt` и добавьте туда файл `persistence.conf`. Как и при форматировании любого диска, соблюдайте осторожность. Если вы отформатируете не тот раздел или не тот диск, то можете потерять что-нибудь важное.

```
# mkfs.ext4 -L persistence /dev/sdb3
mke2fs 1.43-WIP (15-Mar-2016)
Creating filesystem with 7096832 4k blocks and 1777664 inodes
Filesystem UUID: dede20c4-5239-479a-b115-96561ac857b6
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
        4096000

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
# mount /dev/sdb3 /mnt
# echo "/ union" >/mnt/persistence.conf
# ls -l /mnt
total 20
drwx------ 2 root root 16384 May 10 13:31 lost+found
-rw-r--r-- 1 root root     8 May 10 13:34 persistence.conf
# umount /mnt
```

USB-накопитель готов и может быть загружен с помощью пункта меню загрузки Live USB Persistence (Live USB-хранилище).

Создание зашифрованного хранилища на USB-накопителе

Live-boot также поддерживает постоянное хранилище данных на зашифрованном разделе. Такой подход позволяет защитить информацию путем создания зашифрованного раздела LUKS, на котором она и хранится.

Создание зашифрованного хранилища начинается с тех же действий, которые мы выполняли раньше. Однако сейчас вместо форматирования раздела в файловой системе ext4 используйте `cryptsetup` для инициализации раздела в виде LUKS-контейнера. Затем откройте последний и настройте файловую систему ext4 так же, как делали это при создании незашифрованного хранилища, но вместо раздела `/dev/sdb3` примените виртуальный раздел, созданный `cryptsetup`. Этот виртуальный раздел представляет собой расшифрованное содержимое зашифрованного раздела, доступного в `/dev/mapper` под именем, которое вы ему назначили. В нижеприведенном примере мы используем имя `kali_persistence`. И снова убедитесь, что вы форматируете правильный диск и раздел.

```
# cryptsetup --verbose --verify-passphrase luksFormat /dev/sdb3

WARNING!
========
This will overwrite data on /dev/sdb3 irrevocably.

Are you sure? (Type uppercase yes): YES
Enter passphrase:
Verify passphrase:
Command successful.
# cryptsetup luksOpen /dev/sdb3 kali_persistence
Enter passphrase for /dev/sdb3:
# mkfs.ext4 -L persistence /dev/mapper/kali_persistence
mke2fs 1.43-WIP (15-Mar-2016)
Creating filesystem with 7096320 4k blocks and 1774192 inodes
Filesystem UUID: 287892c1-00bb-43cb-b513-81cc9e6fa72b
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
        4096000

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

# mount /dev/mapper/kali_persistence /mnt
# echo "/ union" >/mnt/persistence.conf
# umount /mnt
# cryptsetup luksClose /dev/mapper/kali_persistence
```

Использование нескольких постоянных хранилищ

Если вы пользуетесь вашей live-системой Kali в различных ситуациях, то можете создать несколько файловых систем с разными метками и в командной строке загрузки указывать, какую файловую систему применять в конкретном сеансе работы — для этого служит параметр загрузки `persistence-label=`*`метка`*.

Предположим, вы — профессиональный пентестер. Работая у клиента, вы используете постоянное хранилище, расположенное на зашифрованном разделе. Это нужно для защиты данных на случай, если USB-накопитель украдут или взломают. В то же время вы хотите иметь возможность демонстрировать Kali и какие-нибудь рекламные материалы, хранящиеся на незашифрованном разделе того же носителя. Так как вам не хотелось бы вручную редактировать параметры при каждой загрузке, вы решаете собрать собственный live-образ с отдельными пунктами загрузочного меню.

Первый шаг заключается в сборке собственного live-образа в формате ISO (в соответствии с разделом 9.3, и в частности подразделом «Использование хуков для настройки содержимого образа» текущего раздела). Самое важное — это модифицировать файл `kali-config/common/hooks/live/persistence-menu.binary`, приведя его примерно к такому виду (обратите внимание на параметры `persistence-label`):

```
#!/bin/sh

if [ ! -d isolinux ]; then
    cd binary
fi

cat >>isolinux/live.cfg <<END

label live-demo
    menu label ^Live USB with Demo Data
    linux /live/vmlinuz
    initrd /live/initrd.img
    append boot=live username=root hostname=kali persistence-label=demo
persistence

label live-work
    menu label ^Live USB with Work Data
    linux /live/vmlinuz
    initrd /live/initrd.img
    append boot=live username=root hostname=kali persistence-label=work
      ➥persistence-encryption=luks persistence

END
```

Далее мы собираем ISO-образ и копируем его на USB-накопитель. Затем создаем и инициализируем два раздела и файловые системы, которые будут использоваться для организации постоянных хранилищ информации. Первый раздел создан без шифрования (с меткой `demo`), второй зашифрован (с меткой

work). Исходя из предположения, что USB-накопитель виден в системе как /dev/sdb и размер нашего ISO-образа составляет 3000 Мбайт, нужно выполнить такую последовательность действий:

```
# parted /dev/sdb mkpart primary 3000 MB 55 %
# parted /dev/sdb mkpart primary 55 % 100 %
# mkfs.ext4 -L demo /dev/sdb3
[...]
# mount /dev/sdb3 /mnt
# echo "/ union" >/mnt/persistence.conf
# umount /mnt
# cryptsetup --verbose --verify-passphrase luksFormat /dev/sdb4
[...]
# cryptsetup luksOpen /dev/sdb4 kali_persistence
[...]
# mkfs.ext4 -L work /dev/mapper/kali_persistence
[...]
# mount /dev/mapper/kali_persistence /mnt
# echo "/ union" >/mnt/persistence.conf
# umount /mnt
# cryptsetup luksClose /dev/mapper/kali_persistence
```

Вот и все. Теперь можно загружаться с USB-накопителя и выбирать необходимые пункты из нового загрузочного меню.

Установка пароля самоуничтожения для дополнительной безопасности

В Kali имеется модифицированный cryptsetup для реализации нового свойства: можно установить так называемый пароль самоуничтожения (nuke password), который при использовании уничтожит все ключи, применяемые для работы с зашифрованным разделом.

Это может быть полезным, когда вы много путешествуете и вам нужен быстрый способ обеспечить невозможность доступа к вашим данным. При загрузке просто введите пароль самоуничтожения вместо настоящего пароля, и никто (включая вас) не сможет получить доступ к данным.

Прежде чем использовать эту возможность, весьма полезно будет сделать резервную копию ключей шифрования и сохранить ее в надежном месте.

Если вернуться к приведенному в этом разделе примеру, то, чтобы включить пароль самоуничтожения, можно воспользоваться следующей командой:

```
# cryptsetup luksAddNuke /dev/sdb4
Enter any existing passphrase:
Enter new passphrase for key slot:
Verify passphrase:
```

Более подробную информацию об этой функции можно найти, перейдя по ссылке https://www.kali.org/tutorials/nuke-kali-linux-luks/.

9.5. Резюме

В этой главе вы узнали о способах модификации пакетов с исходным кодом, которые являются основными строительными блоками всех присутствующих в Kali приложений. Кроме того, мы рассказали о том, как настраивать, собирать и устанавливать собственные ядра Kali. Далее мы поговорили об окружении `live-build` и о механизмах создания собственных ISO-образов Kali. Кроме того, в этой главе речь шла о подготовке загрузочных USB-накопителей с поддержкой постоянных хранилищ информации, которые могут быть как зашифрованными, так и нет.

Итоговые сведения по модификации пакетов

Модификацией пакетов Kali обычно занимаются разработчики и те, кто отвечает за поддержку и развитие системы, однако вполне допустима ситуация, что вам будет недостаточно возможностей, предоставляемых официальными сборками пакетов. Поэтому знание того, как собрать модифицированный пакет, может быть очень полезным, особенно если вы хотите поделиться новым пакетом с кем-нибудь, развернуть его на множестве машин или после установки нового пакета без проблем откатить систему к предыдущему состоянию.

Когда вам нужно модифицировать некую программу, весьма соблазнительно скачать исходный код, внести в него изменения и использовать эту модифицированную версию. Однако если программа нуждается в установке, после которой она доступна во всей системе (то есть с помощью `make install`), то такая установка *засорит* файловую систему файлами, неизвестными `dpkg`, что довольно скоро приведет к возникновению проблем, не подлежащих выявлению с помощью анализа зависимостей пакетов. Кроме того, при таком подходе к модификации пакетов результатом сложнее делиться с другими пользователями.

В ходе создания модифицированного пакета общая последовательность действий всегда остается одной и той же: скачивание исходного пакета, извлечение его содержимого, внесение изменений и затем сборка пакета. Существует множество инструментов для решения каждой из этих задач.

Для того чтобы приступить к пересборке пакетов Kali, сначала нужно скачать исходный пакет, включающий основной файл формата `*.dsc` (Debian Source Control) и дополнительные файлы, на которые есть ссылки в основном файле.

Исходные пакеты хранятся на HTTP-зеркалах. Самый эффективный способ скачать их — использовать команду вида `apt source` *`исходное-имя-пакета`*. Для ее успешного выполнения требуется добавить строку `deb-src` в файл `/etc/apt/sources.list` и обновить индексные файлы командой `apt update`.

Кроме того, можно использовать команду `dget` (из пакета devscripts) для непосредственного скачивания файла формата `.dsc` и сопутствующих файлов. В случае с пакетами, подготовленными специально для Kali, исходный код которых хранится в Git-репозитории, загрузить исходный код можно с помощью команды `git clone git://git.kali.org/packages/`*`исходный-пакет`* (если после выполнения такой команды ничего загружено не было, то попытайтесь переключиться на ветку `kali/master` через команду `git checkout kali/master`).

После скачивания исходного кода установите пакеты, перечисленные в зависимостях сборки исходного пакета, с помощью команды `apt build-dep ./`. Ее нужно запустить из каталога пакета, в котором находится исходный код.

Внесение изменений в исходный пакет заключается в выполнении следующих действий.

- Первый шаг состоит в изменении номера версии пакета. Данное действие требуется для того, чтобы система могла отличить новый пакет от исходного. Совершается оно с помощью команды `dch --local` *`идентификатор-версии`* или путем модификации других сведений о пакете с помощью утилиты dch.
- Применение патча с помощью команды `patch -p1 <` *`патч-файл`* или модификация серии патчей quilt.
- Настройка параметров сборки, которые обычно можно найти в файле `debian/rules` или в других файлах из каталога `debian/`.

После модификации исходного пакета можно собрать бинарный пакет с помощью команды `dpkg-buildpackage -us -uc -b`, вызываемой из каталога с исходным кодом. Она создаст неподписанный бинарный пакет, который затем можно установить, воспользовавшись командой `dpkg -i` *`пакет-имя_версии_arch.deb`*.

Итоговые сведения по сборке ядра Linux

У продвинутых пользователей системы иногда возникает потребность в перекомпиляции ядра Kali. Возможно, вы захотите уменьшить размер стандартного ядра, которое по умолчанию содержит большое количество свойств и драйверов, или добавить в него нестандартные драйверы или возможности, или установить патчи ядра. Важно помнить: неправильно настроенное ядро способно сделать систему нестабильной, и вы должны понимать, что команда Kali не может обеспечить обновления безопасности для ядер, которые были собраны пользователями самостоятельно.

В большинстве случаев для внесения изменений в ядро понадобится установить некоторые пакеты с помощью команды `apt install build-essential libncurses5-dev fakeroot`.

Команда `apt-cache search ^linux-source` выведет список последних версий ядра, созданных командой поддержки Kali. Команда `apt install linux-source-`*`номер-версии`* устанавливает сжатый архив с исходным кодом ядра в каталог `/usr/src`.

Исходные файлы должны быть распакованы командой `tar -xaf` в каталог, отличный от `/usr/src` (например, в `~kernel`).

Когда приходит время настраивать ядро, стоит помнить следующее.

- Если вы не являетесь продвинутым пользователем, то сначала стоит заполнить конфигурационный файл ядра. Для этого лучше всего взять стандартную конфигурацию ядра, скопировав `/boot/config-version-string` в `~/kernel/linux-source-version-number/.config`. Как вариант, можно применить команду `make architecture_defconfig` для построения конфигурации, которая подходит для имеющейся архитектуры.

- Инструмент для конфигурирования ядра с текстовым интерфейсом, запускаемый командой `make menuconfig`, считывает файл `.config` и позволяет настраивать ядро с помощью обширной системы меню. При выборе элемента выводится документация по нему и список вероятных значений, и тут же можно ввести для него новое значение.

Выполнение команды `make clean` из каталога с исходным кодом ядра приведет к удалению ранее скомпилированных файлов. Команда `make deb-pkg` создаст до пяти пакетов Debian. Файл `linux-image-version.deb` содержит образ ядра и связанные с ним модули.

Для того чтобы использовать новое ядро, нужно установить необходимые пакеты с помощью команды `dpkg -i file.deb`. При этом нужен пакет `"linux-image"`. Пакет `"linux-headers"` следует установить лишь при наличии внешних модулей ядра для сборки, что происходит, когда у вас есть установленные пакеты `"*-dkms"` (проверить это можно благодаря команде `dpkg -l "*-dkms" | grep ^ii`). Другие пакеты в большинстве случаев не применяются.

Итоговые сведения по сборке собственных ISO-образов Kali

Официальные ISO-образы Kali собраны с помощью набора сценариев live-build, который позволяет полностью автоматизировать и настроить все аспекты создания ISO-образов.

Для использования live-build нужно, чтобы система была обновлена до последней версии.

Конфигурационные данные Kali live-build можно загрузить из Git-репозиториев Kali с помощью команды `apt install curl git live-build`, после которой выполняется команда `git clone git://git.kali.org/live-build-config.git`.

Для создания обновленного, но не модифицированного ISO-образа Kali достаточно воспользоваться командой `./build.sh --verbose`. Сборка займет немало времени, поскольку в ходе выполнения этой операции будут скачаны все необходимые пакеты. После завершения сборки новый ISO-образ можно найти в каталоге `images`. Если при выполнении данной команды применить параметр `--variant` *`вариант`*, то будет собран указанный вариант образа. Различные варианты определяются их каталогами конфигурации `config/variant-*`. Основной образ создается с помощью варианта `gnome`.

Есть несколько способов настройки ISO-образа, которые заключаются во внесении изменений в каталог конфигурации live-build.

- В live-образ можно добавлять пакеты (или удалять существующие), модифицируя файлы `package-lists/*.list.chroot`.
- В образ можно включать собственные пакеты, помещая соответствующие файлы формата `.deb` в каталог `packages.chroot`. Их установка может быть автоматизирована с использованием файлов `preseed/*.cfg`.

- В живую файловую систему можно добавлять файлы, размещая их там, где они должны быть — в каталоге `includes.chroot`.
- В ходе процесса сборки образа с помощью chroot можно выполнять сценарии, устанавливая их как файлы `hooks/live/*.chroot`. Кроме того, сценарии допустимо вызывать во время загрузки с использованием созданного live-образа. Их нужно установить в `/usr/lib/live/config/`*`XXXX-имя`*, например, основываясь на каталоге конфигурации `includes.chroot`.
- Чтобы узнать подробности о конфигурировании и тестировании live-build, можно воспользоваться отличным руководством Debian Live System Manual.

Развернуть ISO-образ Kali на USB-накопителе довольно легко. При этом на таком носителе можно настроить постоянное хранилище информации (как зашифрованное, так и нет). Хотя этот процесс может показаться немного сложным, при ближайшем рассмотрении оказывается, что на переносном носителе совершенно несложно создавать зашифрованные и незашифрованные хранилища. Это значительно расширяет функционал подобных загрузочных носителей.

В следующей главе мы рассмотрим Kali в организации. Обсудим управление конфигурацией и покажем способы расширить и настроить Kali Linux таким образом, чтобы его можно было легко развернуть как на пару, так и на несколько тысяч компьютеров.

ГЛАВА

10

Kali Linux в организации



Ключевые темы:

- установка PXE;
- управление конфигурацией;
- Saltstack;
- разветвление пакетов Kali;
- пакеты конфигурации;
- репозитории пакетов.

К настоящему моменту вы уже убедились, что Kali — чрезвычайно полезный и надежный дистрибутив Debian, обеспечивающий функции промышленной безопасности и шифрования, расширенное управление пакетами, работу на множестве платформ и арсенал инструментов мирового класса для специалистов по безопасности (чем известен в наибольшей степени). Что может быть немного неожиданным, так это то, как Kali масштабируется за пределами рабочего стола до средних и широкомасштабных развертываний и даже до уровня предприятия. В данной главе мы покажем, насколько хорошо Kali может масштабироваться за пределами рабочего стола, обеспечивая централизованное управление и контроль на уровне предприятия на нескольких установках Kali Linux. Короче говоря, после прочтения этой главы вы сможете быстро развертывать высокозащищенные системы Kali, предварительно настроенные для ваших конкретных потребностей, и синхронизировать их благодаря (полуавтоматической) установке обновлений пакетов Kali.

Этот уровень требует нескольких шагов, включая инициирование загрузки компьютера по сети с помощью PXE, применение расширенного инструмента управления конфигурацией (SaltStack), возможность разветвления и настройки пакетов и развертывание хранилища пакетов. Мы подробно рассмотрим каждый шаг, покажем, как избежать «грязной работы», а также развертывать, управлять и поддерживать множество пользовательских установок Kali Linux с относительной легкостью. Помимо всего прочего, мы предоставим толпу миньонов, чтобы помочь вам в управлении вашей империей.

10.1. Установка Kali Linux через сеть (PXE Boot)

Как было рассмотрено в предыдущих главах, стандартный процесс установки Kali Linux прост, стоит только разобраться. Но если вам нужно установить Kali на нескольких компьютерах, то стандартная настройка может оказаться довольно утомительной. К счастью, вы можете запустить процедуру установки Kali, загрузив компьютер через сеть. Это позволяет быстро и легко устанавливать Kali на нескольких машинах одновременно.

Сначала нужно загрузить ваш целевой компьютер по сети. Это делается с помощью Preboot eXecution Environment (PXE) — среды с интерфейсом «клиент-сервер», предназначенной для загрузки любого подключенного к сети компьютера через сеть, даже если на нем нет установленной операционной системы. Для настройки загрузки через сеть с применением PXE требуется настроить по крайней мере сервер TFTP (Trivial File Transfer Protocol — простой протокол передачи файлов) и сервер DHCP/BOOTP (Dynamic Host Configuration Protocol — протокол динамической настройки узла; Bootstrap Protocol — загрузочный протокол). Вам также понадобится веб-сервер, если хотите разместить файл предварительной настройки `debconf`, который будет автоматически использоваться в процессе установки.

К счастью, сервер *dnsmasq* поддерживает как DHCP, так и TFTP, поэтому вам понадобится лишь один сервис, чтобы настроить все необходимое. Кроме того, веб-сервер Apache уже установлен (но не включен) по умолчанию в системе Kali.

Отдельные демоны DHCP и TFTP

Для более сложных настроек набор функций dnsmasq может оказаться несколько ограниченным, или, вероятно, вы захотите включить возможность загрузки с помощью PXE в своей основной сети, которая уже использует DHCP-демон. В обоих случаях придется настроить отдельно демоны DHCP и TFTP.

В руководстве по установке Debian рассмотрена установка isc-dhcp-server и tftpd-hpa для загрузки с помощью PXE (https://www.debian.org/releases/stable/amd64/ch04s05.html).

Прежде чем установить `dnsmasq`, вы должны сначала настроить его в файле `/etc/dnsmasq.conf`. Стандартная настройка содержит лишь несколько ключевых строк:

```
# Network interface to handle
interface=eth0
# DHCP options
# IP range to allocate
dhcp-range=192.168.101.100,192.168.101.200,12h
# Gateway to announce to clients
dhcp-option=option:router,192.168.101.1
# DNS servers to announce to clients
dhcp-option=option:dns-server,8.8.8.8,8.8.4.4
# Boot file to announce to clients
dhcp-boot=pxelinux.0
# TFTP options
enable-tftp
# Directory hosting files to serve
tftp-root=/tftpboot/
```

После настройки файла `/etc/dnsmasq.conf` нужно поместить установочные файлы загрузки в каталог `/tftpboot/`. Для этой цели Kali Linux предоставляет файловый архив, который можно распаковать непосредственно в `/tftpboot/`. Просто выберите между 32-разрядным (i386) и 64-разрядным (amd64) стандартными или графическими (`gtk`) способами установки для вашего целевого компьютера и укажите соответствующий архив:

- http://http.kali.org/dists/kali-rolling/main/installer-amd64/current/images/netboot/gtk/netboot.tar.gz;
- http://http.kali.org/dists/kali-rolling/main/installer-amd64/current/images/netboot/netboot.tar.gz;
- http://http.kali.org/dists/kali-rolling/main/installer-i386/current/images/netboot/gtk/netboot.tar.gz;
- http://http.kali.org/dists/kali-rolling/main/installer-i386/current/images/netboot/netboot.tar.gz.

После того как выбрали архив, создайте каталог `/tftpboot/`, скачайте архив и распакуйте его в этом каталоге:

```
# mkdir /tftpboot
# cd /tftpboot
# wget http://http.kali.org/dists/kali-rolling/main/installer-amd64/current/
    ➥images/netboot/netboot.tar.gz
# tar xf netboot.tar.gz
# ls -l
total 25896
drwxrwxr-x 3 root root     4096 May 6 04:43 debian-installer
lrwxrwxrwx 1 root root       47 May 6 04:43 ldlinux.c32 -> debian-installer/
    ➥amd64/boot-screens/ldlinux.c32
-rw-r--r-- 1 root root 26507247 May 6 04:43 netboot.tar.gz
lrwxrwxrwx 1 root root       33 May 6 04:43 pxelinux.0 -> debian-installer/
    ➥amd64/pxelinux.0
lrwxrwxrwx 1 root root       35 May 6 04:43 pxelinux.cfg -> debian-installer/
    ➥amd64/pxelinux.cfg
-rw-rw-r-- 1 root root       71 May 6 04:43 version.info
```

Распакованные файлы включают загрузчик *pxelinux*, который использует те же файлы конфигурации, что и *syslinux* и *isolinux*. Это позволяет настроить файлы загрузки в `debian-installer/amd64/boot-screens/`, как было бы при создании собственных ISO-образов для Kali Linux.

Например, предположив, что выбрали текстовый установщик, вы можете добавить параметры загрузки для предварительного указания значений языка, страны, раскладки клавиатуры, имени хоста и имени домена. Вы также можете задать установщику внешний URL и настроить таймер, чтобы загрузка выполнялась автоматически, если в течение пяти секунд не была нажата ни одна клавиша. Для этого вы должны сначала изменить файл `debian-installer/amd64/txt.cfg`:

```
label install
        menu label ^Install
        kernel debian-installer/amd64/linux
        append vga=788 initrd=debian-installer/amd64/initrd.
           ➥gz --- quiet language=en country=US keymap=us hostname=kali
           ➥domain= url=http://192.168.101.1/preseed.cfg
```

Затем нужно изменить файл `debian-installer/amd64/syslinux.cfg`, чтобы настроить таймер:

```
# D-I config version 2.0
# search path for the c32 support libraries (libcom32, libutil etc.)
path debian-installer/amd64/boot-screens/
include debian-installer/amd64/boot-screens/menu.cfg
default debian-installer/amd64/boot-screens/vesamenu.c32
prompt 0
timeout 50
```

Располагая средством загружать любую машину через сеть с помощью PXE, вы можете воспользоваться всеми функциями, описанными в разделе 4.3, что позволяет выполнять полную загрузку, пресидинг и автоматическую установку на нескольких компьютерах без физического загрузочного носителя. Не забывайте и о гибкости параметра загрузки `preseed/url=http://`*`сервер`*`/preseed.cfg` (или применении псевдонима `url`), который позволяет установить файл пресидинга для сети.

10.2. Использование управления конфигурацией

Вместе с возможностью быстро устанавливать Kali на нескольких компьютерах сразу вам понадобится помощь в управлении этими машинами после установки. Здесь будут полезны инструменты контроля за конфигурацией для управления машинами или для переключения компьютеров в любое желаемое состояние.

Kali Linux содержит множество популярных инструментов управления конфигурацией, которые вы можете использовать (ansible, chef, puppet, saltstack и т. д.), но в этом разделе мы рассмотрим только `SaltStack` (https://saltstack.com/).

Настройка SaltStack

SaltStack — централизованный сервис управления конфигурацией: salt master управляет множеством миньонов salt minion. Необходимо установить пакеты salt-master на сервер, доступный всем хостам, которыми вы хотите управлять, и salt-minion на этих хостах. Каждому миньону нужно сообщить, где расположен его мастер. Просто отредактируйте файл `/etc/salt/minion` и добавьте ключевое слово `master` в имя DNS (или IP-адрес) Salt-мастера. Обратите внимание, что Salt использует YAML в качестве формата для своих файлов конфигурации.

```
minion# vim /etc/salt/minion
minion# grep ^master /etc/salt/minion
master: 192.168.122.105
```

Каждый миньон имеет уникальный идентификатор, хранящийся в файле `/etc/salt/minion_id`, который по умолчанию соответствует имени его хоста. Этот идентификатор миньона будет использоваться в правилах конфигурации, и потому важно установить его правильно, прежде чем миньон откроет свое соединение с мастером:

```
minion# echo kali-scratch >/etc/salt/minion_id
minion# systemctl enable salt-minion
minion# systemctl start salt-minion
```

Сразу после запуска сервис *salt-minion* попытается связаться с Salt-мастером для обмена определенными криптографическими ключами. Чтобы продолжить соединение, нужно на стороне мастера подтвердить ключ, который предоставил миньон для своей идентификации. Последующие соединения будут устанавливаться автоматически:

```
master# systemctl enable salt-master
master# systemctl start salt-master
master# salt-key --list all
Accepted Keys:
Denied Keys:
Unaccepted Keys:
kali-scratch
Rejected Keys:
master# salt-key --accept kali-scratch
The following keys are going to be accepted:
Unaccepted Keys:
kali-scratch
Proceed? [n/Y] y
Key for minion kali-scratch accepted.
```

Выполнение команд на миньонах

Как только соединение с миньонами установлено, вы можете выполнять на них команды из мастера:

```
master# salt '*' test.ping
kali-scratch:
    True
kali-master:
    True
```

Эта команда приказывает всем миньонам (* — шаблон для обозначения всех миньонов) выполнить функцию `ping` из исполнительного модуля `test`. Данная функция возвращает значение `True` при успешном выполнении и представляет собой простой способ проверить соединение между мастером и различными миньонами.

Вы также можете задать определенный миньон, указав его идентификатор в первом параметре, или множество миньонов, используя более конкретный шаблон (например, `*-scratch` или `kali-*`). Ниже приведен образец выполнения произвольной команды оболочки на миньонах kali-scratch:

```
master# salt kali-scratch cmd.shell 'uptime; uname -a'
kali-scratch:
     05:25:48 up 44 min, 2 users, load average: 0.00, 0.01, 0.05
    Linux kali-scratch 4.5.0-kali1-amd64 #1 SMP Debian 4.5.3-2kali1
      ➥(2016-05-09) x86_64 GNU/Linux
```

Ссылка на Salt-модули Существует множество исполнительных модулей для всех вероятных случаев. Мы не расскажем о них в этой книге, но полный их список доступен по адресу https://docs.saltstack.com/en/latest/ref/modules/all/index.html. Вы также можете получить описание всех исполнительных модулей и их доступных функций на определенном миньоне с помощью команды salt миньон sys.doc. Ее выполнение возвращает очень длинный список функций, но вы можете его отфильтровать, передав имя функции или модуля с его родительским модулем в префиксе в качестве параметра:

```
master# salt kali-scratch sys.doc disk.usage
disk.usage:

    Return usage information for volumes mounted on this minion
```

Один из наиболее полезных модулей — `pkg`, который представляет собой абстрактный менеджер пакетов, полагающийся в своей работе на соответствующий менеджер пакетов для системы (`apt-get` для Debian и его производных, таких как Kali).

Команда `pkg.refresh_db` обновляет список пакетов (то есть выполняется команда `apt-get update`), в то время как команда `pkg.upgrade` устанавливает все доступные обновления (выполняется `apt-get upgrade` или `apt-get dist-upgrade` в зависимости от полученных параметров). Команда `pkg.list_upgrades` покажет список отложенных операций обновления (которые будут выполняться командой `pkg.upgrade dist_upgrade=True`).

Модуль `service` представляет собой абстрактный менеджер сервиса (в случае Kali это `systemd`), который позволяет выполнять все стандартные операции `systemctl`: `service.enable`, `service.disable`, `service.start`, `service.stop`, `service.restart` и `service.reload`:

```
master# salt '*' service.enable ssh
kali-scratch:
    True
kali-master:
    True
master# salt '*' service.start ssh
kali-master:
    True
kali-scratch:
    True
master# salt '*' pkg.refresh_db
kali-scratch:
    ----------
kali-master:
    ----------
master# salt '*' pkg.upgrade dist_upgrade=True
kali-scratch:
```

```
    ----------
    changes:
        ----------
        base-files:
            ----------
            new:
                1:2016.2.1
            old:
                1:2016.2.0
[...]
        zaproxy:
            ----------
            new:
                2.5.0-0kali1
            old:
                2.4.3-0kali3
    comment:
    result:
        True
```

Для более точного примера можно легко настроить сканирование Nmap с помощью dnmap. После установки пакета на всех миньонах следует запустить сервер в первом терминале:

```
server# salt '*' pkg.install dnmap
[...]
server# vim dnmap.txt
server# dnmap_server -f dnmap.txt
```

Предполагая, что IP-адрес сервера 1.2.3.4, вы можете приказать всем миньонам запустить процесс на стороне клиента, который подключается к серверу:

```
server# salt '*' cmd.run_bg template=jinja 'dnmap_client -s 1.2.3.4 -a {{
    ➥grains.id }}'
kali-scratch:
    ----------
    pid:
        17137
[...]
```

Обратите внимание: в примере используется cmd.run_bg для запуска команды dnmap_client в фоновом режиме. Не ждите, пока он закончится, так как это длительный процесс. К сожалению, он не завершается должным образом, когда вы прерываете работу сервера, поэтому вам придется очистить его самостоятельно:

```
server# salt '*' cmd.shell 'pkill -f dnmap_client'
```

State-файлы salt и другие особенности

Хотя удаленное выполнение команд — важный элемент, это лишь малая часть того, что умеет SaltStack.

При настройке новой машины часто приходится запускать множество команд и тестов для задания более детальной конфигурации системы перед установкой. Эти операции могут быть формализованы в шаблонах конфигурации многоразового использования, называемых файлами *state* (статус). Операции, описанные в state-файлах, выполняются с помощью команды `state.apply salt`.

Чтобы сэкономить время, можете рассчитывать на множество готовых к использованию state-файлов, которые были созданы сообществом Kali и которые можно найти в «Salt-формулах» (https://docs.saltstack.com/en/latest/topics/development/conventions/formulas.html):

Существует множество функций SaltStack, которые можно использовать комбинированно:

- запланированное выполнение действий;
- определение действий в ответ на события, вызванные миньонами;
- сбор данных из миньонов;
- координация последовательности операций на нескольких миньонах;
- применение статусов по SSH без установки сервиса salt-minion;
- системы предоставления облачных инфраструктур и управления ими и др.

Функциональность SaltStack довольно обширна, и мы не можем раскрыть здесь все свойства. Но есть книги, посвященные исключительно SaltStack, а также исчерпывающая онлайн-документация. Если хотите узнать больше о возможностях SaltStack, то следуйте по этой ссылке: https://docs.saltstack.com/en/latest/.

Если вы управляете большим количеством компьютеров, то вам будет полезно узнать больше о SaltStack, так как он позволит сэкономить время при подключении новых машин и вы сможете поддерживать согласованную конфигурацию в вашей сети.

Чтобы помочь вам понять суть работы со state-файлами, мы рассмотрим простой пример: как активировать хранилище APT и установить пакет, который вы создадите в подразделах «Создание хранилища пакетов для APT» и «Создание пакетов конфигурации» раздела 10.3. Мы также разберем регистрацию SSH-ключа в учетной записи `root`, чтобы в случае возникновения проблем вы могли войти в систему удаленно.

По умолчанию state-файлы хранятся в каталоге `/srv/salt` на устройстве мастера; это структурированные файлы YAML с расширением `.sls`. Как и для запуска команд, применение статусов зависит от многих state-модулей:

- https://docs.saltstack.com/en/latest/topics/tutorials/starting_states.html;
- https://docs.saltstack.com/en/latest/ref/states/all/.

Ваш файл `/srv/salt/offsec.sls` вызовет три модуля:

```
offsec_repository:
  pkgrepo.managed:
    - name: deb http://pkgrepo.offsec.com offsec-internal main
```

```
    - file: /etc/apt/sources.list.d/offsec.list
    - key_url: salt://offsec-apt-key.asc
    - require_in:
      - pkg: offsec-defaults

offsec-defaults:
  pkg.installed

ssh_key_for_root:
  ssh_auth.present:
    - user: root
    - name: ssh-rsa AAAAB3NzaC1yc2...89C4N rhertzog@kali
```

Статус `offsec_repository` полагается на state-модуль `pkgrepo`. В примере для регистрации хранилища пакетов используется функция `managed` в этом state-модуле. С помощью параметра `key_url` вы даете salt знать, что GPG-ключ (защищенный ASCII), необходимый для проверки подписи хранилища, может быть получен из файла `/srv/salt/offsec-aptkey.asc` на salt-мастере. Параметр `require_in` гарантирует, что этот статус будет обработан до использования `offsec-defaults`, так как последний требует правильно настроенное хранилище для установки пакета.

Статус `offsec-defaults` устанавливает одноименный пакет. Это говорит о том, что имя ключа часто является важным значением для статусов, хотя его всегда можно переопределить с помощью параметра `name` (как было сделано для предыдущего статуса). Для простых случаев, подобных приведенному, это понятно и лаконично.

Последний статус (`ssh_key_for_root`) добавляет ключ SSH, указанный в параметре `name`, в `/root/.ssh/ authorized_keys` (притом целевой пользователь указан параметром `user`). Обратите внимание: в данном примере мы сократили ключ для простоты чтения, но вы должны поместить ключ в параметр `name` целиком.

Далее этот state-файл можно применить к заданному миньону:

```
server# salt kali-scratch state.apply offsec
kali-scratch:
----------
          ID: offsec_repository
    Function: pkgrepo.managed
        Name: deb http://pkgrepo.offsec.com offsec-internal main
      Result: True
     Comment: Configured package repo 'deb http://pkgrepo.offsec.com
              ➥offsec-internal main'
     Started: 06:00:15.767794
    Duration: 4707.35 ms
     Changes:
              ----------
              repo:
                  deb http://pkgrepo.offsec.com offsec-internal main
```

```
----------
          ID: offsec-defaults
    Function: pkg.installed
      Result: True
     Comment: The following packages were installed/updated: offsec-defaults
     Started: 06:00:21.325184
    Duration: 19246.041 ms
     Changes:
              ----------
              offsec-defaults:
                  ----------
                  new:
                      1.0
                  old:
----------
          ID: ssh_key_for_root
    Function: ssh_auth.present
        Name: ssh-rsa AAAAB3NzaC1yc2...89C4N rhertzog@kali
      Result: True
     Comment: The authorized host key AAAAB3NzaC1yc2...89C4N for user root
              ➥was added
     Started: 06:00:40.582539
    Duration: 62.103 ms
     Changes:
              ----------
              AAAAB3NzaC1yc2...89C4N:
                  New
Summary for kali-scratch
------------
Succeeded: 3 (changed=3)
Failed: 0
------------
Total states run:     3
Total run time:  24.015 s
```

Он также может быть неизменно связан с миньоном, будучи записанным в файле /srv/salt/top.sls, который используется командой state.highstate для применения всех соответствующих статусов за один подход:

```
server# cat /srv/salt/top.sls
base:
  kali-scratch:
    - offsec
server# salt kali-scratch state.highstate
kali-scratch:
----------
          ID: offsec_repository
    Function: pkgrepo.managed
        Name: deb http://pkgrepo.offsec.com offsec-internal main
      Result: True
```

```
     Comment: Package repo 'deb http://pkgrepo.offsec.com offsec-internal
              ➥main' already configured
     Started: 06:06:20.650053
    Duration: 62.805 ms
     Changes:
----------
          ID: offsec-defaults
    Function: pkg.installed
      Result: True
     Comment: Package offsec-defaults is already installed
     Started: 06:06:21.436193
    Duration: 385.092 ms
     Changes:
----------
          ID: ssh_key_for_root
    Function: ssh_auth.present
        Name: ssh-rsa AAAAB3NzaC1yc2...89C4N rhertzog@kali
      Result: True
     Comment: The authorized host key AAAAB3NzaC1yc2...89C4N is already
              ➥present for user root
     Started: 06:06:21.821811
    Duration: 1.936 ms
     Changes:
Summary for kali-scratch
------------
Succeeded: 3
Failed:    0
------------
Total states run:     3
Total run time: 449.833 ms
```

10.3. Расширение и настройка Kali Linux

Иногда вам требуется изменить настройки дистрибутива, чтобы работа системы соответствовала вашим нуждам. Лучший способ добиться этого — создать собственное хранилище пакетов, включающее модифицированные версии пакетов Kali, которые вам пришлось разветвить, а также дополнительные пакеты, предоставляющие настраиваемую конфигурацию и дополнительное программное обеспечение (не от Kali Linux).

Разветвление пакетов Kali

Пожалуйста, обратитесь к разделу 9.1 для разъяснений этой темы.

Все пакеты можно разветвить, если у вас есть на то веская причина. Но знайте, что разветвление имеет свою цену, так как вам придется обновлять его каждый

раз, когда Kali публикует обновление. Ниже представлено несколько возможных причин для разветвления пакета.

- Добавление патча, чтобы исправить ошибку или включить новую функцию. Хотя в большинстве случаев лучше отправить этот патч разработчикам, чтобы ошибка была исправлена или функция была добавлена прямо в исходный код.
- Компиляция пакета с различными параметрами (при условии, что есть веские причины, по которым команда Kali не скомпилировала их изначально с этими параметрами; иначе стоило бы обратиться к разработчикам Kali с просьбой включить нужные параметры).

В противовес к сказанному выше перечислим некоторые причины, по которым не стоит разветвлять пакет, а также варианты решения вашей проблемы.

- Изменение файла конфигурации. Существует множество лучших вариантов, таких как использование управления конфигурацией для автоматической установки измененного файла конфигурации или установка пакета конфигурации, который поместит файл в каталог конфигурации (если она доступна) или перенаправит исходный файл конфигурации.
- Обновление до новой официальной версии. Опять же, лучше работать с разработчиками для обновления пакета непосредственно в Debian или Kali. С моделью скользящего релиза обновления достигают конечных пользователей достаточно быстро.

Среди всех доступных пакетов есть те, которые являются одними из основных элементов Kali Linux и которые может быть интересно разветвить в ряде ситуаций:

- kali-meta: этот исходный пакет создает все метапакеты kali-linux-*, и в частности `kali-linux-full`, определяющий, какие пакеты установлены в стандартном ISO-образе Kali Linux;
- desktop-base: этот исходный пакет содержит различные файлы, которые используются по умолчанию для настольных установок. Рассмотрите возможность разветвления данного пакета, если хотите демонстрировать бренд вашей организации в фоновом режиме по умолчанию или изменить тему рабочего стола;
- kali-menu: этот пакет определяет структуру меню Kali и предоставляет файлы `.desktop` для всех приложений, которые должны быть указаны в меню Kali.

Создание пакетов конфигурации

Теперь, когда мы рассмотрели загрузку с помощью PXE и обсудили управление конфигурацией с применением SaltStack, а также разветвление пакетов, настало время использовать все это на практике и создать собственный пакет конфигурации для полуавтоматического развертывания настраиваемой конфигурации одновременно на нескольких машинах.

В данном примере вы создадите настраиваемый пакет, который настраивает и задействует ваше собственное хранилище пакетов и ключ подписи GnuPG, распределяет конфигурацию SaltStack, устанавливает пользовательский фон и предоставляет стандартные настройки рабочего стола по умолчанию для всех ваших установок Kali.

Это может показаться сложной задачей (особенно если вы пролистали руководство Debian New Maintainer Guide), но, к счастью для нас, пакет конфигурации обычно представляет собой современный файловый архив, который довольно просто превращается в пакет.

> **Пример пакета** Если вам любопытно взглянуть на настоящий пакет, который по сути является пакетом конфигурации, то рассмотрите пакет kali-defaults. Он не так прост, как образец из этого раздела, но обладает всеми соответствующими характеристиками и даже использует ряд расширенных методов (например, dpkg-divert) для замены файлов, предоставленных ранее другими пакетами.

Пакет offsec-defaults будет содержать в себе несколько файлов:

- `/etc/apt/sources.list.d/offsec.list`: запись `source.list` для APT, активирующая внутреннее хранилище пакетов компании;
- `/etc/apt/trusted.gpg.d/offsec.gpg`: ключ GnuPG, используемый для подписи внутреннего хранилища пакетов компании;
- `/etc/salt/minion.d/offsec.conf`: файл конфигурации SaltStack, указывающий, где расположен Salt-мастер;
- `/usr/share/images/offsec/background.png`: качественное фоновое изображение с логотипом Offensive Security;
- `/usr/share/glib-2.0/schemas/90_offsec-defaults.gschema.override`: файл, предоставляющий альтернативные настройки по умолчанию для рабочего стола GNOME.

Сначала создайте каталог `offsec-defaults-1.0` и поместите в него все перечисленные файлы. Затем выполните команду `dh_make --native` (из пакета dh-make), чтобы добавить инструкции по пакетированию Debian, которые будут храниться в подкаталоге `debian`:

```
$ mkdir offsec-defaults-1.0; cd offsec-defaults-1.0
$ dh_make --native
Type of package: (single, indep, library, python)
[s/i/l/p]? i
Email-Address       : buxy@kali.org
License             : gpl3
Package Name        : offsec-defaults
Maintainer Name     : Raphaël Hertzog
Version             : 1.0
```

```
Package Type      : indep
Date              : Thu, 16 Jun 2016 18:04:21 +0200
Are the details correct? [Y/n/q] y
Currently there is not top level Makefile. This may require additional tuning
Done. Please edit the files in the debian/ subdirectory now.
```

Сначала вам предлагают выбрать тип пакета. В данном примере мы выбрали тип `indep`; он указывает на то, что этот исходный пакет сгенерирует единственный двоичный пакет, который может быть общим для всех архитектур (`Architecture: all`). Тип `single` создает один двоичный пакет, зависящий от целевой архитектуры (`Architecture: any`). В таком случае тип `indep` более уместен, поскольку пакет содержит только текстовые файлы и не включает двоичных программ, поэтому его можно использовать на компьютерах всех архитектур. Тип `library` применим к общим библиотекам ввиду того, что они должны следовать строгим правилам пакетирования. Аналогично тип `python` нужно ограничить модулями Python.

Имя разработчика и адрес электронной почты

Большинство программ, связанных с обслуживанием пакетов, будут искать ваше имя и адрес электронной почты в переменных среды DEBFULLNAME и DEBEMAIL или EMAIL. Достаточно определить их один раз и навсегда, и это предотвратит их неоднократный повторный ввод. Если вы обычно используете оболочку Bash, то это лишь вопрос добавления следующих двух строк в файл ~/.bashrc:

```
export EMAIL="buxy@kali.org"
export DEBFULLNAME="Raphael Hertzog"
```

Команда `dh_make` создает подкаталог `debian`, содержащий множество файлов. Некоторые из них необходимы, в частности `rules`, `control`, `changelog` и `copyright`. Файлы с расширением `.ex` — это образцы файлов, которые можно использовать, изменив их содержимое и удаляя данное расширение. Когда они больше не нужны, мы рекомендуем удалить их. Файл `compat` нужно хранить, поскольку он требуется для правильной работы набора программ *debhelper* (все они начинаются с префикса `dh_`), который применяется на разных этапах процесса сборки пакета.

Файл `copyright` должен содержать информацию об авторах документов, входящих в пакет, и соответствующую лицензию. Если лицензия по умолчанию, выбранная `dh_make`, вам не подходит, то необходимо отредактировать этот файл. Ниже представлена измененная версия файла `copyright` (авторского права):

```
Format: https://www.debian.org/doc/packaging-manuals/copyright-format/1.0/
Upstream-Name: offsec-defaults

Files: *
Copyright: 2016 Offensive Security
License: GPL-3.0+

License: GPL-3.0+
 This program is free software: you can redistribute it and/or modify
```

```
 it under the terms of the GNU General Public License as published by
 the Free Software Foundation, either version 3 of the License, or
 (at your option) any later version.
 .
 This package is distributed in the hope that it will be useful,
 but WITHOUT ANY WARRANTY; without even the implied warranty of
 MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 GNU General Public License for more details.
 .
 You should have received a copy of the GNU General Public License
 along with this program. If not, see <https://www.gnu.org/licenses/>.
 .
 On Debian systems, the complete text of the GNU General
 Public License version 3 can be found in "/usr/share/common-licenses/GPL-3".
```

Стандартный файл `changelog` обычно не требует изменений; достаточно заменить строку `Initial release` более подробным объяснением:

```
offsec-defaults (1.0) unstable; urgency=medium

  * Add salt minion's configuration file.
  * Add an APT's sources.list entry and an APT's trusted GPG key.
  * Override the gsettings schema defining the background picture.

 -- Raphaël Hertzog <buxy@kali.org> Thu, 16 Jun 2016 18:04:21 +0200
```

Внесем изменения в файл `control`. Мы изменим поле `Section` на *`misc`* и удалим поля `Homepage`, `Vcs-Git` и `Vcs-Browser`. Наконец, заполним поле `Description`:

```
Source: offsec-defaults
Section: misc
Priority: optional
Maintainer: Raphaël Hertzog <buxy@kali.org>
Build-Depends: debhelper (>= 9)
Standards-Version: 3.9.8

Package: offsec-defaults
Architecture: all
Depends: ${misc:Depends}
Description: Default settings for Offensive Security
 This package contains multiple files to configure computers
 owned by Offensive Security.
 .
 It notably modifies:
  - APT's configuration
  - salt-minion's configuration
  - the default desktop settings
```

Файл `rules` обычно содержит набор правил, используемых для настройки, сборки и установки программного обеспечения в определенном подкаталоге (названном аналогично сгенерированному двоичному пакету). Содержимое этого подкаталога затем архивируется в пакете Debian, как если бы это был корень файловой системы.

В таком случае файлы будут установлены в подкаталоге `debian/offsec-defaults/`. Например, чтобы завершить установку пакета `/etc/apt/sources.list.d/offsec.list`, установите файл в `debian/offsec-defaults/etc/apt/sources.list.d/offsec.list`. Файл `rules` служит в качестве файла `Makefile` с несколькими стандартными параметрами (включая `clean` и `binary`, используемыми для очистки исходного каталога и создания двоичного пакета соответственно).

Что такое файл Makefile?

Возможно, вы заметили сообщение о недостающем файле Makefile в конце программного вывода dh_make и упоминании его сходства с файлом rules. Makefile — это сценарий, используемый программой make; он описывает правила создания набора файлов друг от друга в дереве зависимостей. Например, программа может быть собрана из набора исходных файлов. Файл Makefile описывает эти правила в следующем формате:

```
target: source1 source2 ...
        command1
        command2
```

Значение такого правила заключается в следующем: если один из source*-файлов более поздний, чем файл target (целевой), то последний должен быть сгенерирован с помощью команд command1 и command2.

Обратите внимание: командные строки должны начинаться с символа табуляции; также заметьте, что когда командная строка начинается с символа тире (—), сбой этой команды не прерывает весь процесс.

Хотя данный файл является основой процесса, он содержит лишь минимум для запуска стандартного набора команд, предоставляемых инструментом `debhelper`. Это относится к файлам, созданным `dh_make`. Для установки большинства файлов мы рекомендуем настроить поведение команды `dh_install`, создав такой файл `debian/offsec-defaults.install`:

```
apt/offsec.list etc/apt/sources.list.d/
apt/offsec.gpg etc/apt/trusted.gpg.d/
salt/offsec.conf etc/salt/minion.d/
images/background.png usr/share/images/offsec/
```

Вы также можете использовать данный прием для установки файла переопределения `gsettings`, но `debhelper` предоставляет для этого специальный инструмент (`dh_installgsettings`), так что удобнее применить его. Для начала укажите свои настройки в файле `debian/offsec-defaults.gsettings-override`:

```
[org.gnome.desktop.background]
picture-options='zoom'
picture-uri='file:///usr/share/images/offsec/background.png'
```

Затем переопределите вызов dh_installgsettings в debian/rules, чтобы увеличить приоритет до уровня, необходимого для переопределения организации (90 согласно руководству):

```
#!/usr/bin/make -f

 %:
        dh $@
override_dh_installgsettings:
           dh_installgsettings --priority=90
```

Итак, исходный пакет готов. Остается лишь создать двоичный пакет с помощью уже знакомого вам метода: запустите команду dpkg-buildpackage -us -uc из каталога offsec-defaults-1.0:

```
$ dpkg-buildpackage -us -uc
dpkg-buildpackage: info: source package offsec-defaults
dpkg-buildpackage: info: source version 1.0
dpkg-buildpackage: info: source distribution unstable
dpkg-buildpackage: info: source changed by Raphaël Hertzog <buxy@kali.org>
dpkg-buildpackage: info: host architecture amd64
 dpkg-source --before-build offsec-defaults-1.0
 fakeroot debian/rules clean
dh clean
   dh_testdir
   dh_auto_clean
   dh_clean
dpkg-source -b offsec-defaults-1.0
dpkg-source: info: using source format '3.0 (native)'
dpkg-source: info: building offsec-defaults in offsec-defaults_1.0.tar.xz
dpkg-source: info: building offsec-defaults in offsec-defaults_1.0.dsc
 debian/rules build
dh build
   dh_testdir
   dh_update_autotools_config
   dh_auto_configure
   dh_auto_build
   dh_auto_test
 fakeroot debian/rules binary
dh binary
   dh_testroot
   dh_prep
   dh_auto_install
   dh_install
   dh_installdocs
   dh_installchangelogs
   debian/rules override_dh_installgsettings
make[1]: Entering directory '/home/rhertzog/kali/kali-book/samples/
   ➥offsec-defaults-1.0'
dh_installgsettings --priority=90
```

```
make[1]: Leaving directory '/home/rhertzog/kali/kali-book/samples/offsec-
   ➥defaults-1.0'
   dh_perl
   dh_link
   dh_strip_nondeterminism
   dh_compress
   dh_fixperms
   dh_installdeb
   dh_gencontrol
   dh_md5sums
   dh_builddeb
dpkg-deb: building package 'offsec-defaults' in
   ➥'../offsec-defaults_1.0_all.deb'.
 dpkg-genchanges >../offsec-defaults_1.0_amd64.changes
dpkg-genchanges: info: including full source code in upload
 dpkg-source --after-build offsec-defaults-1.0
dpkg-buildpackage: info: full upload; Debian-native package (full source
   ➥is included)
```

Создание хранилища пакетов для APT

Теперь, имея собственный пакет, вы можете распространить его через хранилище пакетов APT. Используйте инструмент `reprepro` для создания желаемого репозитория и его наполнения. Это довольно мощный инструмент, и, безусловно, стоит сначала ознакомиться с руководством по нему.

Хранилище пакетов обычно размещается на сервере. Чтобы отделить его от других сервисов, запущенных на сервере, лучше всего создать отдельного пользователя специально для этого сервиса. В данной учетной записи пользователя вы сможете размещать файлы хранилища, а также ключ GnuPG, который будет служить для подписи хранилища пакетов:

```
# apt install reprepro gnupg
[...]
# adduser --system --group pkgrepo
Adding system user 'pkgrepo' (UID 136) ...
Adding new group 'pkgrepo' (GID 142) ...
Adding new user 'pkgrepo' (UID 136) with group 'pkgrepo' ...
Creating home directory '/home/pkgrepo' ...
# chown pkgrepo $(tty)
# su - -s /bin/bash pkgrepo
$ gpg --gen-key
gpg (GnuPG) 2.1.11; Copyright (C) 2016 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

gpg: directory '/home/pkgrepo/.gnupg' created
gpg: new configuration file '/home/pkgrepo/.gnupg/dirmngr.conf' created
gpg: new configuration file '/home/pkgrepo/.gnupg/gpg.conf' created
```

```
gpg: keybox '/home/pkgrepo/.gnupg/pubring.kbx' created
Note: Use "gpg --full-gen-key" for a full featured key generation dialog.

GnuPG needs to construct a user ID to identify your key.

Real name: Offensive Security Repository Signing Key
Email address: repoadmin@offsec.com
You selected this USER-ID:
    "Offensive Security Repository Signing Key <repoadmin@offsec.com>"
Change (N)ame, (E)mail, or (O)kay/(Q)uit? o
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
[...]
gpg: /home/pkgrepo/.gnupg/trustdb.gpg: trustdb created
gpg: key B4EF2D0D marked as ultimately trusted
gpg: directory '/home/pkgrepo/.gnupg/openpgp-revocs.d' created
gpg: revocation certificate stored as '/home/pkgrepo/.gnupg/openpgp-revocs.d/
     ➥F8FE22F74F1B714E38DA6181B27F74F7B4EF2D0D.rev'
public and secret key created and signed.

gpg: checking the trustdb
gpg: marginals needed: 3 completes needed: 1 trust model: PGP
gpg: depth: 0 valid: 1 signed: 0 trust: 0-, 0q, 0n, 0m, 0f, 1u
pub   rsa2048/B4EF2D0D 2016-06-17 [S]
      Key fingerprint = F8FE 22F7 4F1B 714E 38DA 6181 B27F 74F7 B4EF 2D0D
uid         [ultimate] Offensive Security Repository Signing Key
     ➥<repoadmin@offsec.com>
sub   rsa2048/38035F38 2016-06-17 []
```

Обратите внимание: когда вам будет предложено ввести пароль, вы должны ввести пустое значение (и подтвердить, что не хотите защищать свой ключ), поскольку заинтересованы в неинтерактивном подписании репозитория. Кроме того, заметьте, что gpg требует доступ к терминалу с правом на запись с целью иметь возможность безопасно запрашивать пароль: именно поэтому необходимо было изменить владельца виртуального терминала (который принадлежит root, так как вы первоначально были подключены на правах данного пользователя), прежде чем запускать оболочку pkgrepo.

Теперь можно начать настройку хранилища. Для reprepro необходимо создать специальный каталог и внутри него создать файл conf/distributions, указывающий, какие дистрибутивы доступны в хранилище пакетов:

```
$ mkdir -p reprepro/conf
$ cd reprepro
$ cat >conf/distributions <<END
Codename: offsec-internal
AlsoAcceptFor: unstable
Origin: Offensive Security
```

```
Description: Offsec's Internal packages
Architectures: source amd64 i386
Components: main
SignWith: F8FE22F74F1B714E38DA6181B27F74F7B4EF2D0D
END
```

Обязательные поля — `Codename` (название), в котором указывается имя дистрибутива; `Architectures` (архитектуры), определяющее, какие архитектуры будут доступны в дистрибутиве (и приняты на стороне ввода); `Components` (компоненты), которое указывает на различные компоненты, доступные в дистрибутиве (последние — своего рода подразделения дистрибутива, которые могут быть включены отдельно через APT-файл `sources.list`). Поля `Origin` (происхождение) и `Description` (описание) являются чисто информативными и преимущественно копируются из файла `Release` (релиз). Поле `SignWith` просит `reprepro` подписать хранилище с помощью ключа GnuPG, идентификатор которого был указан (применяйте здесь полную цифровую подпись, чтобы убедиться в использовании правильного ключа и отстутствии совпадений в короткой форме идентификатора с любым другим ключом). Настройка `AlsoAcceptFor` не обязательна, но делает возможной обработку файлов формата `.changes`, значение поля `Distribution` которых указано здесь (без этого оно будет принимать только кодовое имя дистрибутива в данном поле).

С помощью этой базовой настройки вы позволите `reprepro` сгенерировать пустой репозиторий:

```
$ reprepro export
Exporting indices...
$ find .
.
./db
./db/version
./db/references.db
./db/contents.cache.db
./db/checksums.db
./db/packages.db
./db/release.caches.db
./conf
./conf/distributions
./dists
./dists/offsec-internal
./dists/offsec-internal/Release.gpg
./dists/offsec-internal/Release
./dists/offsec-internal/main
./dists/offsec-internal/main/source
./dists/offsec-internal/main/source/Release
./dists/offsec-internal/main/source/Sources.gz
./dists/offsec-internal/main/binary-amd64
./dists/offsec-internal/main/binary-amd64/Packages
./dists/offsec-internal/main/binary-amd64/Release
```

```
./dists/offsec-internal/main/binary-amd64/Packages.gz
./dists/offsec-internal/main/binary-i386
./dists/offsec-internal/main/binary-i386/Packages
./dists/offsec-internal/main/binary-i386/Release
./dists/offsec-internal/main/binary-i386/Packages.gz
./dists/offsec-internal/InRelease
```

Как видите, reprepro создал метаданные хранилища в подкаталоге dists. Он также создал внутреннюю базу данных в подкаталоге db.

Настало время поместить в хранилище ваш первый пакет. Сначала скопируйте файлы, сгенерированные при сборке пакета offsec-defaults (offsec-defaults_1.0.dsc, offsec-defaults_1.0.tar.xz, offsec-defaults_1.0_all.deb и offsec-defaults_1.0_amd64.changes), в katаталог /tmp на сервере, на котором размещен репозиторий пакетов, и дайте указание reprepro включить пакет:

```
$ reprepro include offsec-internal /tmp/offsec-defaults_1.0_amd64.changes
Exporting indices...
$ find pool
pool
pool/main
pool/main/o
pool/main/o/offsec-defaults
pool/main/o/offsec-defaults/offsec-defaults_1.0.dsc
pool/main/o/offsec-defaults/offsec-defaults_1.0.tar.xz
pool/main/o/offsec-defaults/offsec-defaults_1.0_all.deb
```

Как вы можете видеть, он добавил файлы в собственный пул пакетов в подкаталоге pool.

Каталоги dists и pool — это два каталога, которые необходимо сделать (обще-)доступными через HTTP, чтобы завершить настройку вашего хранилища APT. Они содержат в себе все файлы, которые могут понадобиться APT для скачивания.

Предположим, вы хотите разместить пакеты на виртуальном хосте с именем pkgrepo.offsec.com. Для этого можно создать следующий файл конфигурации Apache, сохранить его в /etc/apache2/sites-available/pkgrepo.offsec.com.conf и активировать его с помощью команды a2ensite pkgrepo.offsec.com:

```
<VirtualHost *:80>
    ServerName pkgrepo.offsec.com
    ServerAdmin repoadmin@offsec.com

    ErrorLog /var/log/apache2/pkgrepo.offsec.com-error.log
    CustomLog /var/log/apache2/pkgrepo.offsec.com-access.log "%h %l %u %t
        ➥\"%r\" %>s %O"

    DocumentRoot /home/pkgrepo/reprepro

    <Directory "/home/pkgrepo/reprepro">
        Options Indexes FollowSymLinks MultiViews
        Require all granted
```

```
        AllowOverride All
    </Directory>
</VirtualHost>
```

На машинах, которые будут использовать пакеты из этого хранилища, нужно добавить следующую запись в файл `sources.list`:

```
deb http://pkgrepo.offsec.com offsec-internal main

# Enable next line if you want access to source packages too
# deb-src http://pkgrepo.offsec.com offsec-internal main
```

Теперь ваш пакет размещен в хранилище и должен быть доступен для ваших сетевых хостов.

Хотя процесс установки был длительным и утомительным, на этом вся «грязная работа» окончена. Вы можете загружать компьютеры вашей сети через PXE, устанавливать свою версию Kali Linux удаленно благодаря предварительным настройкам, передаваемым через сеть, настраивать SaltStack для управления вашими конфигурациями (и контроля миньонов!), создавать разветвленные пакеты и распространять их через собственное хранилище пакетов. Все это обеспечивает централизованное управление и контроль на уровне предприятия с помощью одновременно нескольких установок дистрибутива. Короче говоря, теперь вы можете быстро развертывать высокозащищенные системы Kali, предварительно настроенные для ваших конкретных потребностей, и синхронизировать их благодаря (полуавтоматической) установке обновлений пакетов Kali.

10.4. Резюме

Kali Linux легко масштабируется за пределами рабочего стола до средних и широкомасштабных развертываний и даже до уровня предприятия. В этой главе мы рассмотрели, как централизовать управление несколькими установками Kali с помощью SaltStack, что позволяет быстро развертывать высокозащищенные системы Kali, предварительно настроенные для ваших конкретных нужд. Мы также продемонстрировали, как их синхронизировать благодаря (полуавтоматической) установке обновлений пакетов Kali.

Мы обсудили разветвление пакетов, позволяющее создавать собственные настраиваемые исходные пакеты, которыми можно делиться с другими.

Кратко рассмотрим основные шаги, необходимые для создания Salt-мастеров и миньонов, которые позволяют осуществлять дистанционное управление и настройку удаленных хостов.

- Загрузите компьютер через сеть с помощью PXE, с настроенным файловым сервером TFTP, сервером DHCP/BOOTP (и веб-сервером для предварительной настройки debconf). Сервер `dnsmasq` поддерживает как DHCP, так и TFTP, а веб-сервер `apache2` уже установлен (но не включен) по умолчанию в системе Kali.

- В руководстве по установке Debian рассмотрена настройка `isc-dhcp-server` и `tftpd-hpa` для загрузчика через PXE (https://www.debian.org/releases/stable/amd64/ch04s05.html).
- Сервер `dnsmasq` настраивается через файл `/etc/dnsmasq.conf`. Стандартная настройка включает лишь несколько ключевых строк:

```
# Network interface to handle
interface=eth0
# DHCP options
# IP range to allocate
dhcp-range=192.168.101.100,192.168.101.200,12h
# Gateway to announce to clients
dhcp-option=option:router,192.168.101.1
# DNS servers to announce to clients
dhcp-option=option:dns-server,8.8.8.8,8.8.4.4
# Boot file to announce to clients
dhcp-boot=pxelinux.0
# TFTP options
enable-tftp
# Directory hosting files to serve
tftp-root=/tftpboot/
```

- Распакуйте 32-разрядные (i386), 64-разрядные (amd64), стандартные или графические (gtk) установочные файлы загрузки из архива Kali в каталог `/tftpboot/`. Архивы можно найти здесь:
  - http://http.kali.org/dists/kali-rolling/main/installer-amd64/current/images/netboot/gtk/netboot.tar.gz;
  - http://http.kali.org/dists/kali-rolling/main/installer-amd64/current/images/netboot/netboot.tar.gz;
  - http://http.kali.org/dists/kali-rolling/main/installer-i386/current/images/netboot/gtk/netboot.tar.gz;
  - http://http.kali.org/dists/kali-rolling/main/installer-i386/current/images/netboot/netboot.tar.gz.

```
# mkdir /tftpboot
# cd /tftpboot
# wget http://http.kali.org/dists/kali-rolling/main/installer-amd64/
    ➥current/images/netboot/netboot.tar.gz
# tar xf netboot.tar.gz
```

- При необходимости внесите изменения в файл `txt.cfg` для предварительной настройки параметров или установки таймеров (см. раздел 4.3). Затем вы можете использовать инструменты управления конфигурацией для контроля за машинами или переключения удаленных компьютеров в любое желаемое состояние.
- SaltStack — это централизованный сервис управления конфигурацией: Salt-мастер управляет множеством Salt-миньонов. Установите пакет salt-master на доступном сервере и salt-minion на управляемых компьютерах.

- Отредактируйте файл конфигурации `/etc/salt/minion`, который хранится в формате YAML, и добавьте ключевое слово `master` в имя DNS (или IP-адрес) Salt-мастера.
- Установите уникальный идентификатор миньона в `/etc/salt/minion_id`:

```
minion# echo kali-scratch >/etc/salt/minion_id
minion# systemctl enable salt-minion
minion# systemctl start salt-minion
```

- После этого произойдет обмен ключами. На стороне мастера подтвердите ключ идентификации миньона. Последующие соединения будут устанавливаться автоматически:

```
master# systemctl enable salt-master
master# systemctl start salt-master
master# salt-key --list all
Accepted Keys:
Denied Keys:
Unaccepted Keys:
kali-scratch
Rejected Keys:
master# salt-key --accept kali-scratch
The following keys are going to be accepted:
Unaccepted Keys:
kali-scratch
Proceed? [n/Y] y
Key for minion kali-scratch accepted.
```

- После подключения миньонов вы можете выполнять на них команды мастера. Примеры:

```
master# salt '*' test.ping
kali-scratch:
True
kali-master:
True
master# salt kali-scratch cmd.shell 'uptime; uname -a'
master# salt kali-scratch sys.doc'
master# salt '*' service.enable ssh
[...]
master# salt '*' service.start ssh
[...]
master# salt '*' pkg.refresh_db
[...]
master# salt '*' pkg.upgrade dist_upgrade=True
server# salt '*' cmd.shell 'pkill -f dnmap_client'
```

- Полный список исполнительных модулей можно найти по адресу https://docs.saltstack.com/en/latest/ref/modules/all/index.html.
- Применяйте файлы SaltState (шаблоны конфигурации многоразового использования), чтобы планировать действия, собирать данные, координировать

последовательности операций на нескольких миньонах, предоставлять облачные инфраструктуры, управлять ими и др. Сэкономьте свое время с помощью готовых к использованию «Salt-формул» (https://docs.saltstack.com/en/latest/topics/development/conventions/formulas.html).

- Когда вы решаете разветвить пакет, сначала определите, так ли уж это необходимо. Разветвление пакетов имеет значительные преимущества и недостатки. Внимательно ознакомьтесь с ними. Неплохим выбором для разветвления могут стать пакеты `kali-meta`, `desktop-base` и `kali-menu`. Процесс разветвления пакета довольно трудоемкий, и его сложно описать кратко.

Теперь, когда мы изучили все основы касательно установки, настройки и развертывания Kali Linux, рассмотрим его роль в области информационной безопасности.

ГЛАВА 11

Оценка защищенности информационных систем



Ключевые темы:

- типы оценок;
- оценка уязвимости;
- соответствие тестирования на проникновение;
- традиционные тестирования на проникновение;
- оценка приложения;
- типы атак;
- отказ обслуживания;
- повреждение памяти;
- веб-уязвимости;
- взлом паролей;
- атаки на клиента.

К этому моменту мы рассмотрели немало возможностей Kali Linux, поэтому вы уже должны хорошо понимать особенности системы и то, как с ее помощью решать множество сложных задач.

Однако, прежде чем приступать к использованию Kali на практике, стоит разобраться с некоторыми понятиями, связанными с оценкой защищенности информационных систем. В данной главе мы познакомим вас с этими понятиями и предоставим ссылки на дополнительные материалы, которые пригодятся в том случае, если вам понадобится задействовать Kali для выполнения оценки защищенности.

Для начала стоит уделить время самому понятию «безопасность» применительно к информационным системам. При попытках защитить информационную систему обычно обращают внимание на три ее основных атрибута:

- *конфиденциальность* (confidentiality): могут ли лица, у которых не должно быть доступа к системе или информации, получить его?
- *целостность* (integrity): можно ли несанкционированно модифицировать систему или данные?
- *доступность* (availability): можно ли нормально, учитывая время и способ доступа, пользоваться системой или данными?

Все вместе эти понятия составляют так называемую модель CIA (confidentiality, integrity, availability) и во многом являются основными элементами, которым уделяют внимание при защите систем в ходе стандартных процессов развертывания, поддержки или оценки защищенности.

Кроме того, полезно отметить, что в ряде случаев одни аспекты CIA будут заботить вас больше, чем другие. Например, у вас есть личный дневник, который содержит ваши самые заветные мысли. Конфиденциальность этой информации может быть гораздо важнее, чем ее целостность или доступность. Другими словами, вас не должно сильно беспокоить, если кто-то что-то напишет в него, не читая. Точно так же вам не нужно, чтобы дневник был абсолютно всегда под рукой. С другой стороны, при защите системы, которая хранит информацию о выписанных медицинских рецептах, целостность данных выходит на первый план. Важно не дать посторонним читать эти записи (то есть получать сведения о том, кто какие лекарства принимает), как и обеспечить постоянную доступность списка рецептов. Но главнее всего — отслеживать, чтобы никто не смог изменить содержимое системы (то есть повлиять на ее целостность), поскольку это может привести к опасным для жизни последствиям.

Когда вы занимаетесь безопасностью системы и обнаруживаете проблему, вам нужно понять, какие части CIA или их комбинации имеют отношение к данной проблеме. Это помогает понять ее более полно, позволяет разбивать инциденты по категориям и принимать соответствующие меры. Понимая сущность модели CIA, несложно классифицировать с ее помощью уязвимости разного масштаба.



Например, можно рассмотреть сквозь призму CIA взлом веб-приложения через внедрение SQL-кода (описано ниже).

- *Конфиденциальность*: взлом с помощью SQL-инъекции позволяет нарушителю извлечь содержимое веб-приложения, открывает полный доступ на чтение всех данных, но не дает возможности изменять информацию или нарушать работоспособность базы данных.
- *Целостность*: взлом с помощью SQL-инъекции дает злоумышленнику возможность изменять информацию, которая уже имеется в базе данных. Он не может читать данные или блокировать доступ к базе.
- *Доступность*: взлом с помощью SQL-инъекции позволяет инициировать тяжелый запрос, потребляющий большой объем серверных ресурсов. Несколько таких запросов приводят к отказу сервиса (DoS-атака). У злоумышленника нет возможности читать или изменять данные, но он может помешать обычным пользователям работать с веб-приложением.
- *Множественные угрозы*: взлом с помощью SQL-инъекции дает полный доступ к операционной системе сервера, на котором выполняется веб-приложение. Обладая таким доступом, атакующий может нарушить конфиденциальность системы, получив доступ к любым нужным ему данным, нарушить целостность системы, изменяя данные, и при желании нарушить работоспособность веб-приложения, что приведет к недоступности системы для обычных пользователей.

Концепции модели CIA довольно просты, и на самом деле вы уже интуитивно пользуетесь ими, сами того не подозревая. Однако важно осмысленно применять модель, поскольку она способна помочь понять, в какую сторону стоит направить усилия. Эта концептуальная база позволит определить критически важные компоненты системы, а также объем усилий и ресурсов, которые стоит вложить в устранение обнаруженных проблем.

Кроме того, мы рассмотрим подробнее *риск* и его составляющие: *угрозу* и *уязвимость*. Эти понятия не слишком сложны, но в них легко запутаться. Более детально мы разберем их позже, но если сказать об этом в двух словах, то можно отметить следующее: лучше всего воспринимать *риск* как то, что вы пытаетесь предотвратить, *угрозу* — как того, кто может это, нежелательное, совершить, и *уязвимость* — как нечто, способное позволить сделать то, что вы хотите предотвратить. Соответствующие усилия можно приложить к снижению уровня угрозы или устранения уязвимости, тем самым уменьшить уровень риска.

Например, путешествуя по миру, вы можете подвергнуться значительному *риску* заражения малярией. Это потому, что в некоторых местностях высока *угроза* быть укушенным малярийным комаром и у вас, почти наверняка, нет иммунитета к малярии. К счастью, вы можете контролировать вашу *уязвимость* с помощью медикаментов и попытаться контролировать *угрозу*, используя репелленты и противомоскитные сетки. Целенаправленно контролируя и *угрозу*, и *уязвимость*, вы можете предотвратить *риск* заражения.

11.1. Kali Linux в оценке защищенности

Если вы готовитесь использовать Kali Linux в бою, то сначала необходимо удостовериться в том, что у вас установлена чистая и рабочая система. Распространенная ошибка многих начинающих пентестеров заключается в использовании одного и того же экземпляра Kali в ходе анализа защищенности разных систем. Такой подход может привести к проблемам по двум основным причинам.

- В ходе оценки часто приходится вручную устанавливать, настраивать или каким-либо другим образом менять систему. Эти единичные изменения могут помочь быстрее привести Kali в рабочее состояние или решить конкретную проблему, но их тяжело контролировать; они усложняют поддержку системы и ее дальнейшую настройку.
- Каждая задача по оценке безопасности системы уникальна. Поэтому использование системы, в которой остались, скажем, заметки, код и другие изменения после анализа системы одного клиента, у другого клиента может привести к путанице и к тому, что данные клиентов окажутся перемешанными.

Именно поэтому настоятельно рекомендуем начинать работу с чистой установки Kali, а усилия на подготовку предварительно настроенной версии Kali Linux, которая готова к автоматической установке, быстро окупаются. Для того чтобы обзавестись подобной версией системы, обратитесь к разделам 9.3 и 4.3. Чем серьезнее вы подойдете к автоматизации своей работы сегодня, тем меньше времени потратите завтра.

У каждого свои требования касательно настройки Kali Linux, но есть ряд универсальных рекомендаций, на которые стоит обратить внимание всем. Для начала рассмотрите возможность использования зашифрованной установки, как показано в подразделе «Установка на полностью зашифрованную файловую систему» раздела 4.2. Это позволит защитить ваши данные, хранящиеся на компьютере, и может быть чрезвычайно полезно, если ваш ноутбук будет украден.

Для обеспечения дополнительной безопасности во время путешествий имеет смысл рассмотреть настройку функции самоуничтожения (см. врезку «Установка пароля самоуничтожения для дополнительной безопасности» в главе 9) после отправки (зашифрованной) копии ключа коллеге в офисе. Таким образом, ваши данные будут защищены до тех пор, пока вы не вернетесь в офис, где сможете восстановить работоспособность компьютера с помощью ключа дешифрования.

Еще одна вещь, к которой стоит отнестись предельно серьезно, — это какие именно пакеты вы установили. Готовясь к выполнению очередного задания, обратите внимание на то, что из инструментов пригодится. Например, собираясь приняться за поиск дыр в беспроводной сети, вы можете рассмотреть вероятность установки метапакета kali-linux-wireless, содержащего все средства исследования беспроводных сетей, доступные в Kali Linux. Готовясь к испытаниям веб-приложения, вы можете подготовить все инструменты, предназначенные для подобных задач, установив метапакет kali-linux-web. Лучше всего исходить из предположения, что

во время тестирования у вас не будет нормального доступа в Интернет, поэтому нужно как можно лучше подготовиться заранее.

По той же причине вам, вероятно, следует перепроверить сетевые настройки (см. разделы 5.1 и 7.3). Дважды проверьте настройки DHCP и просмотрите сервисы, которые слушают ваш IP-адрес. Эти установки могут оказать серьезнейшее воздействие на успешность работы. Вы не можете анализировать то, что не видите, а излишние сервисы способны выдать вашу систему и привести к ее отключению от сети еще до начала оценки.

Если вы занимаетесь расследованием сетевых вторжений, то будьте особенно внимательны к вашим сетевым настройкам и избегайте любого воздействия на системы, которые подверглись атаке. Специально настроенная версия Kali с метапакетом kali-linux-forensic загружается в режиме криминалистической экспертизы (forensic mode). В нем система не монтирует диски автоматически и не использует раздел подкачки. В результате при использовании инструментов цифровой криминалистики, доступных в Kali, вы сможете сохранить целостность анализируемой системы.

В заключение можно сказать, что правильная подготовка Kali Linux к работе, использование чистой, вдумчиво настроенной системы — это залог вашего успеха.

11.2. Типы оценок

После того как вы убедились в том, что ваша система Kali готова к работе, пришло время точно определить, какое именно исследование вы собираетесь провести. В целом можно выделить четыре вида подобных исследований: *оценка уязвимости*, *оценка систем на соответствие стандартам безопасности*, *традиционное тестирование на проникновение* и *оценка приложений*. Исследование системы может включать различные элементы каждого вида, и стоит рассказать о них подробнее, раскрыв их связь с Kali Linux и рабочим окружением.

Прежде чем переходить к описанию конкретных видов мероприятий по оценке безопасности, сначала разберемся, чем уязвимости отличаются от эксплойтов.

Уязвимость можно определить как дефект информационной системы, позволяющий нарушить ее конфиденциальность, целостность или доступность. Существуют различные виды потенциальных уязвимостей. Ниже представлены некоторые из них.

- **Включение файлов** (https://en.wikipedia.org/wiki/File_inclusion_vulnerability): уязвимость ко включению файлов в веб-приложениях позволяет вам *включать* локальные или удаленные файлы в процесс выполняемых программой вычислений. Например, у веб-приложения может быть функция «Сообщение дня», которая читает содержимое некоего файла и включает его в веб-страницу для демонстрации пользователям. Если при разработке подобной функции были допущены ошибки, то она может позволить злоумышленнику изменять выполняемый им запрос к сайту таким образом, чтобы веб-приложение вместо нужного файла подключило к странице файл, выбранный злоумышленником.

- **SQL-инъекция** (https://en.wikipedia.org/wiki/SQL_injection): атака приложения методом SQL-инъекции представляет собой ситуацию, когда нарушитель обходит подсистемы проверки данных, вводимых пользователями. Это позволяет ему вводить собственные SQL-команды, которые будет выполнять взломанная система. Подобное может привести к проблемам в области безопасности.
- **Переполнения буфера** (https://en.wikipedia.org/wiki/Buffer_overflow): это уязвимость, позволяющая злоумышленнику обойти подсистемы контроля ввода и записать данные в область памяти, которая соседствует с памятью, выделенной для определенного буфера. В ряде случаев области памяти рядом с буфером могут быть чрезвычайно важны для обеспечения работоспособности атакуемой программы, и правильная манипуляция данными в этих областях позволяет получить контроль над выполнением кода.
- **Состояние гонки** (https://en.wikipedia.org/wiki/Race_condition): это уязвимость, основанная на использовании временных зависимостей в программах. В некоторых случаях рабочий процесс приложения зависит от выполнения определенной последовательности действий (событий). Ее изменение может привести к уязвимости.

С другой стороны, *эксплойт* — это программа, которая в случае ее применения может воспользоваться конкретной уязвимостью, хотя и не все уязвимости пригодны к использованию таким образом. Поскольку эксплойт должен изменить выполняющийся процесс, заставляя его совершать нежелательные действия, то создание эксплойтов может оказаться сложной задачей. Более того, в современных вычислительных платформах имеется целый ряд технологий противодействия эксплойтам, усложняющих их разработку. Среди таких технологий — предотвращение выполнения данных (Data Execution Prevention, DEP) (https://en.wikipedia.org/wiki/Executable_space_protection#Windows) и рандомизация размещения адресного пространства (Address Space Layout Randomization, ASLR) (https://en.wikipedia.org/wiki/Address_space_layout_randomization). Однако если для некоей уязвимости не удается обнаружить общедоступный эксплойт, то это еще не значит, что он не существует или его нельзя создать. Например, многие организации продают специально разработанные эксплойты, которые никогда не станут общедоступными, поэтому все уязвимости следует рассматривать как потенциально подверженные эксплуатации.

Оценка уязвимости систем

Уязвимость — это дефект информационной системы, который может быть каким-то образом использован для нарушения ее конфиденциальности, целостности или доступности. При проведении оценки уязвимости систем главное — создать перечень уязвимостей, обнаруженных в *целевом окружении*. Понятие «целевое окружение» очень важно при оценке защищенности систем. Вы не должны проводить работы за пределами целевой сети клиента или отклоняться от задач исследования. Выход за пределы исследуемого окружения может привести к нарушению работы сервисов клиента, к потере его доверия или судебному иску против вас и вашего работодателя.

Исследование уязвимостей систем благодаря его простоте часто выполняется на регулярной основе в достаточно зрелых окружениях в рамках демонстрации уровня их защищенности или соответствия неким стандартам безопасности. В большинстве случаев автоматические инструменты наподобие тех, что можно найти в категориях Vulnerability Analysis (Анализ уязвимостей) (https://tools.kali.org/category/vulnerability-analysis) и Web Applications (Веб-приложения) (https://tools.kali.org/category/web-applications) на сайте Kali Tools или в соответствующих разделах меню Applications (Приложения) на рабочем столе Kali, используются для обнаружения live-систем в целевом окружении, идентификации сервисов, прослушивающих некие порты, и их анализа. Делается это для сбора как можно большего количества информации о системе, такой как сведения о программном обеспечении серверов, версиях, платформах и т. д.

Затем собранную информацию проверяют на известные сигнатуры уязвимостей. Последние состоят из комбинаций фрагментов данных, которые позволяют распознать известные проблемы в области безопасности. Здесь используется как можно больше сведений, поскольку чем больше их будет, тем точнее окажется идентификация уязвимости. Существует множество показателей, представляющих интерес при анализе уязвимостей систем. Среди них можно отметить следующие.

- *Версия операционной системы*: нередко бывает так, что некое приложение уязвимо в одной версии ОС, но не в другой. Именно поэтому сканер попытается определить максимально точно, на какой версии операционной системы работает целевое приложение.
- *Уровень патча*: нередко выходят патчи для операционных систем, при установке которых версия ОС не меняется, хотя известные уязвимости при этом либо исчезают, либо начинают вести себя не так, как раньше.
- *Архитектура процессора*: есть множество приложений, доступных для разных архитектур процессора, таких как Intel x86 и Intel x64, для различных версий ARM, для UltraSPARC и т. д. В некоторых случаях уязвимость существует только на конкретной архитектуре процессора, поэтому знание последней может сыграть важную роль при формировании точной сигнатуры уязвимости.
- *Версия целевого программного обеспечения*: это один из основных показателей, который нужно получить для идентификации уязвимости.

После сбора информации эти и многие другие показатели будут использованы для формирования сигнатур уязвимостей. Вполне ожидаемо, что чем больше фрагментов данных совпадут с сигнатурой, тем увереннее можно говорить об обнаруженной уязвимости. Занимаясь сопоставлением собранной информации с сигнатурами, можно получить разные результаты.

- *Положительный результат*: сигнатура соответствует профилю уязвимости, уязвимость в системе обнаружена. Получив подобный результат, нужно заняться источником проблемы и исправить уязвимость, так как это именно то, чем может воспользоваться злоумышленник, чтобы навредить вашей организации (или организации вашего клиента).

- *Ложноположительный результат*: в ходе анализа удалось обнаружить совпадение с сигнатурой уязвимости. Однако то, что найдено, уязвимостью не является. Подобные результаты обычно считают «информационным шумом», они усложняют работу. Для более четкого разделения истинных и мнимых уязвимостей требуется более глубокий анализ ситуации.
- *Отрицательный результат*: после сканирования системы совпадений с известными сигнатурами уязвимостей обнаружить не удалось, следовательно, уязвимости в системе нет. Это идеальный сценарий, доказывающий отсутствие известных уязвимостей в целевой системе.
- *Ложноотрицательный результат*: совпадения с сигнатурами найти не удалось, но уязвимость в системе имеется. Получение ложноположительных результатов — это плохо, но ложноотрицательный результат — еще хуже. В подобной ситуации проблема существует, но сканер ее не находит, и поэтому у вас нет показателей ее наличия.

Несложно понять, что для обеспечения надежных результатов сканирования системы чрезвычайно важна точность сигнатур. Чем больше данных удается собрать, тем выше качество результатов автоматического сканирования, основанного на сигнатурах. Именно поэтому весьма популярно сканирование с предварительной аутентификацией в системе.

При таком подходе сканирующее программное обеспечение использует предоставленные специалисту данные для аутентификации в целевой системе. Это позволяет анализировать информацию, недоступную в других условиях. Например, при обычном сканировании реально получить лишь сведения о системе, которые можно извлечь из анализа сервисов, доступных извне, и из анализа предоставленного ими функционала. Иногда и такое сканирование позволяет собрать немало данных. Но эти сведения не идут ни в какое сравнение с тем, что можно узнать о системе, войдя в нее и тщательно проанализировав все установленное ПО, примененные патчи, исполняющиеся процессы и т. д. Сбор столь обширных данных о системе полезен для нахождения уязвимостей, которые в противном случае могли бы остаться ненайденными.

Хорошо проведенное исследование уязвимостей дает отчет о потенциальных проблемах и показатели, которые можно использовать для анализа изменения ситуации во времени. Такое исследование довольно простое, но, даже учитывая это, многие организации регулярно проводят автоматическое сканирование уязвимостей. Делается это обычно в часы минимальной нагрузки на системы, поскольку сканирование может потребовать немалых сетевых и серверных ресурсов и помешать обычной работе организации.

Как уже было сказано, в ходе сканирования систем на наличие уязвимостей необходимо проверить множество различных фрагментов данных, чтобы получить точные результаты. Все эти проверки могут создать нагрузку на целевую систему и сеть. К сожалению, сложно заранее узнать, сколько ресурсов будет потреблено, поскольку это зависит от количества запущенных сервисов и типов проверок, свя-

занных с последними. Сканирование потребляет системные ресурсы, поэтому при использовании соответствующих инструментов важно представлять себе, какую нагрузку они могут создавать на исследуемые системы и сети.

Многопоточное сканирование	Большинство сканеров уязвимостей поддерживают настройку количества потоков сканирования, то есть сколько проверок будет выполняться одновременно. Увеличение количества потоков сканирования приводит к росту нагрузки на исследуемую платформу (компьютер), сеть, целевые системы. Заманчиво ускорить сканирование за счет увеличения количества потоков, но важно помнить о том, что это может привести к значительному росту нагрузки на системы.

Когда проверка завершена, обнаруженные уязвимости обычно связывают со стандартными идентификаторами, такими как номера CVE (https://cve.mitre.org/), EDB-ID или коды классификации уязвимостей, принятые у поставщиков инструментов сканирования. Данная информация вместе со сведениями об оценке уязвимостей по методике CVSS (https://www.first.org/cvss/) используется для определения уровня риска. Все эти сведения с учетом ложноположительных и ложноотрицательных сообщений об уязвимостях дают общее представление об уязвимостях, которые нужно учитывать, анализируя результаты сканирования.

Поскольку автоматизированные средства используют для выявления уязвимостей базы данных сигнатур, то малейшее отклонение от известной сигнатуры способно изменить результат и, соответственно, обоснованность сообщений об обнаруженных уязвимостях. Ложноположительные результаты указывают на то, чего нет, а ложноотрицательные, наоборот, скрывают существующие проблемы. Поэтому качество и возможности автоматических сканеров уязвимости напрямую зависят от применяемых ими баз данных сигнатур. Как правило, поставщики подобного ПО предлагают разные версии своих программ. Некоторые бесплатные, снабженные урезанными базами, предназначены для домашних пользователей. Другие же довольно дорогие, с полноценными базами, обычно ориентированы на корпоративных пользователей.

Еще одна проблема, которая часто возникает при сканировании на наличие уязвимостей, заключается в пригодности к использованию предложенных рейтингов риска. Последние определяют на универсальной основе, принимая во внимание множество различных факторов, таких как уровень привилегий, тип программного обеспечения, возможность реализации уязвимости до или после аутентификации. Подобные рейтинги нельзя применять слепо, поскольку их ценность зависит от особенностей исследуемой системы. Качественно оценить уровень риска можно только при использовании тех рейтингов, которые основаны на анализе подробной информации о системе и обнаруженных в ней уязвимостях.

Нет единого общепринятого соглашения о рейтингах рисков, однако можно порекомендовать взять за основу оценки рисков в исследуемой среде стандарт NIST

SP 800-30 (https://csrc.nist.gov/publications/search?requestserieslist=1&requeststatuslist=1,3&requestdisplayoption=brief&itemsperpage=all&requestsortorder=5#800-30). Данный стандарт определяет реальный риск обнаруженной уязвимости как *комбинацию возможности осуществления угрозы и уровня воздействия на организацию при ее воплощении*.

Вероятность осуществления угрозы

В соответствии с NIST (National Institute of Standards and Technology — Национальный институт стандартов и технологий) возможность осуществления угрозы основана на вероятности того, что источник угрозы способен к эксплуатации конкретной уязвимости с вероятностью, которая соответствует одному из рейтингов: низкому, среднему или высокому.

- Высокий: потенциальный противник высококвалифицирован и мотивирован, меры, принятые для защиты от использования уязвимости, недостаточны.
- Средний: потенциальный противник мотивирован и квалифицирован, но меры по защите от применения уязвимости могут препятствовать его успеху.
- Низкий: потенциальный противник неквалифицирован или испытывает недостаток мотивации, при этом приняты меры для защиты от использования уязвимости, которые частично или абсолютно эффективны.

Воздействие

Уровень воздействия на организацию при осуществлении угрозы определяют, оценивая размер ущерба, который может причинить использование анализируемой уязвимости.

- Высокий: использование уязвимости способно привести к значительным финансовым потерям, серьезно повредить миссии или репутации организации, может закончиться серьезным материальным ущербом или человеческими жертвами.
- Средний: применение уязвимости способно привести к финансовым потерям, повредить миссии или репутации компании, или, в случае с человеческими ресурсами компании, привести к травмам.
- Низкий: использование уязвимости может привести к некоторым финансовым потерям или воздействию на миссию или репутацию компании.

Общий риск

Точное знание возможности осуществления угрозы и уровня ее воздействия на организацию позволяет оценить уровень риска, который выражается в виде функции от двух найденных показателей. Показатель уровня риска позволяет сформировать план действий для тех, кто отвечает за защиту и поддержку анализируемой системы.

- Высокий: имеется серьезная потребность в принятии дополнительных мер для защиты от уязвимости. В некоторых случаях можно позволить системе про-

должать работу, однако нужно подготовить план по ее защите, который следует реализовать как можно скорее.

- Средний: есть потребность в принятии дополнительных мер для защиты от уязвимости. План по реализации необходимых мер по защите должен быть выполнен в пределах разумного периода времени.
- Низкий: владелец системы определит самостоятельно, реализовывать ли дополнительные меры по защите от уязвимости либо принять обнаруженный риск и оставить систему неизменной.

Итоговые мероприятия

Показатель риска обнаруженной уязвимости формирует множество факторов, поэтому рейтинги, полученные из систем автоматического сканирования, следует рассматривать лишь как отправную точку в определении реального риска.

По результатам оценки уязвимостей составляют отчеты. Подобные отчеты, созданные со знанием дела и профессионально проанализированные, закладывают базу для других исследований, таких как оценка систем на соответствие стандартам безопасности. Важно извлечь из результатов оценки уязвимостей все возможное.

Kali создает отличную платформу для проведения оценки уязвимостей, которая не нуждается в особой настройке. В разделах меню Applications (Приложения) можно найти множество инструментов для анализа уязвимостей. В частности, речь идет о разделах Information Gathering (Сбор информации), Vulnerability Analysis (Анализ уязвимостей) и Web Application Analysis (Анализ веб-приложения). Узнать подробности об использовании дистрибутива для анализа уязвимостей можно на сайте Kali Linux Tools Listing (https://tools.kali.org/tools-listing), на сайте Kali Linux Official Documentation (https://docs.kali.org/) и ознакомившись с бесплатным курсом Metasploit Unleashed (https://www.offensive-security.com/metasploit-unleashed/).

Оценка систем на соответствие стандартам безопасности

Следующий по сложности вид проверки — это оценка систем на соответствие стандартам безопасности. Подобные испытания систем наиболее часты, так как основаны на анализе требований, предписываемых государственными и индустриальными стандартами, распространяющимися на организации.

Существует множество специализированных стандартов безопасности, однако чаще всего встречается стандарт безопасности данных индустрии платежных карт (Payment Card Industry Data Security Standard, PCI DSS) (https://www.pcisecuritystandards.org/documents/Penetration_Testing_Guidance_March_2015.pdf). Его создали компании, выпускающие платежные карты. Ему должны соответствовать организации, обрабатывающие карточные платежи. Если говорить о других распространенных стандартах, то можно отметить такие как Defense Information Systems Agency Security Technical Implementation Guides (DISA STIG) (https://iase.disa.mil/stigs/Pages/index.aspx),

Federal Risk and Authorization Management Program (FedRAMP) (https://www.fedramp.gov/about-us/about/), Federal Information Security Management Act (FISMA) (https://csrc.nist.gov/projects/risk-management) и др.

Корпоративный клиент может заказать подобную проверку или обратиться за результатами ранее проведенного исследования по разным причинам. Независимо от того, являются ли они обязательными или выполняются по инициативе клиента, такие исследования называют «оценкой систем на соответствие стандартам безопасности», или «исследованиями на соответствие стандартам безопасности», или «проверками на соответствие стандартам безопасности».

Оценка системы на соответствие стандартам обычно начинается с анализа уязвимостей. В случае с процедурой проведения аудита на соответствие стандарту PCI (https://www.pcisecuritystandards.org/documents/PCIDSS_QRGv3_2.pdf) оценка уязвимостей, если она проведена соответствующим образом, может удовлетворить нескольким основным требованиям стандарта, в том числе: «2. Не использовать пароли и другие системные параметры, заданные производителем по умолчанию» (можно применить утилиты из категории меню Password Attack (Взлом паролей)), «11. Регулярно выполнять тестирование систем и процессов обеспечения безопасности» (с помощью инструментов из категории Database Assessment (Исследование баз данных)) и др. Некоторые требования нельзя проверить с помощью обычных средств сканирования на уязвимости. Среди них: «9. Ограничить физический доступ к данным держателей карт» и «12. Разработать и поддерживать политику информационной безопасности для всего персонала организации». Для проверки таких требований нужны дополнительные усилия.

На первый взгляд может показаться не вполне понятным, как использовать Kali Linux для выполнения некоторых проверок. Однако Kali отлично подходит для решения подобных задач, причем не только из-за обширного набора стандартных инструментов, но и потому, что он основана на Debian, что открывает возможность установки множества дополнительных приложений. Искать программы, реализующие необходимый функционал, можно в менеджере пакетов с помощью ключевых слов, взятых из используемого стандарта информационной безопасности. Подобный поиск почти наверняка завершится выдачей нескольких заслуживающих внимания результатов. В настоящее время многие организации используют Kali Linux как платформу именно для оценки систем на соответствие стандартам безопасности.

Традиционное тестирование на проникновение

В последнее время стало сложно подобрать подходящее определение для «традиционного теста на проникновение». Дело в том, что подобные тесты используются в различных сферах деятельности, и, как результат, все описывают их по-своему. Путаницы добавляет и то, что «тестированием на проникновение» все чаще стали называть описанную выше оценку систем на соответствие стандартам безопасности или даже обычную оценку уязвимостей. В подобных случаях проверка не выходит за рамки неких минимальных требований.

В этом подразделе мы не будем касаться споров об особенностях различных видов испытаний систем. Здесь мы расскажем об исследованиях, не ограниченных некими «минимальными требованиями». Это исследования, которые задуманы так, чтобы после их проведения можно было по-настоящему улучшить общую безопасность организации.

В противоположность видам исследований, которые мы обсудили ранее, традиционные тесты на проникновение не часто начинаются с определения области проверки. Вместо этого для них устанавливают определенные цели. Например: «смоделировать последствия компрометации внутреннего пользователя» или «выяснить, что случилось бы, если бы организация попала под целенаправленную атаку, выполняемую внешним злоумышленником». Ключевой отличительной чертой подобного анализа является то, что в ходе его выполнения не только находят и оценивают уязвимости, но еще и используют найденные проблемы для раскрытия наихудших вариантов развития событий. В ходе тестирования на проникновение не полагаются исключительно на инструменты сканирования систем на уязвимости. Работа продолжается с помощью исследования находок, применения эксплойтов или испытаний для исключения ложноположительных результатов, делается все возможное для обнаружения скрытых уязвимостей или того, что мы называем ложноотрицательными результатами. Подобное исследование часто включает эксплуатацию обнаруженных уязвимостей, оценку уровня доступа, который предоставляют эксплойты, и использование этого повышенного уровня доступа как отправной точки для дополнительных атак на целевую систему.

Здесь требуется критический анализ целевого окружения, ручной поиск уязвимостей, креативность, способность к нестандартному мышлению. Все это — подходы к обнаружению дополнительных уязвимостей, которые требуют других инструментов, способных найти уязвимости там, где заканчиваются возможности самых мощных автоматических сканеров. Нередко после завершения этого шага весь процесс начинают снова для того, чтобы обеспечить полное и качественное выполнение работы.

Несмотря на сложность и многоплановость традиционного тестирования на проникновение, ход такого исследования можно упорядочить, разбив на несколько шагов. Стоит отметить, что Kali упрощает подбор программы для каждого из таких шагов с помощью Kali Menu.

- *Сбор информации*: на данном шаге усилия пентестера направлены на то, чтобы узнать как можно больше о целевом окружении. Обычно его деятельность неинвазивна и выглядит как рядовая активность пользователей. Эти действия составляют основу для остальных этапов исследования и таким образом должны привести к сбору как можно более полных данных о системе. Раздел Information Gathering (Сбор информации) в меню Kali Linux содержит в себе десятки инструментов, направленных на то, чтобы раскрыть как можно больше сведений об исследуемой системе.
- *Обнаружение уязвимостей*: этот шаг часто называют «активным сбором информации». Специалист, пытаясь идентифицировать потенциальные уязвимости

в целевом окружении, еще не атакует систему, но уже ведет себя не так, как обычный пользователь. Именно здесь часто имеет место вышеописанное сканирование систем на уязвимости. На данном шаге исследования будут полезны программы из разделов Vulnerability Analysis (Анализ уязвимостей), Web Application Analysis (Анализ веб-приложений), Database Assessment (Исследование баз данных) и Reverse Engineering (Обратное проектирование).

- *Эксплуатация уязвимостей*: имея список обнаруженных потенциальных уязвимостей, на данном шаге исследования специалист пытается воспользоваться ими для того, чтобы найти точку опоры в целевой среде. В этом деле полезны инструменты, которые можно найти в категориях Web Application Analysis (Анализ веб-приложений), Database Assessment (Исследование баз данных), Password Attacks (Взлом паролей) и Exploitation Tools (Средства эксплуатации уязвимостей).
- *Проникновение и извлечение данных*: после того как исследователю удалось закрепиться в системе, нужно двигаться дальше. Как правило, на этом шаге ищут способ повысить привилегии до уровня, соответствующего тому, который необходим для достижения целевых систем, ранее недоступных, и скрытного извлечения из них секретной информации. Здесь можно обратиться к таким разделам меню приложений, как Password Attacks (Взлом паролей), Exploitation Tools (Средства эксплуатации уязвимостей), Sniffing & Spoofing (Сниффинг и спуфинг), и Post Exploitation (Действия после эксплуатации уязвимости).
- *Подготовка отчетов*: после завершения активной фазы исследования нужно задокументировать произведенные действия и подготовить отчет. Обычно этот шаг не отличается такой же технической сложностью, как предыдущие. Однако благодаря качественным отчетам клиент способен получить полную отдачу от проделанной работы, так что не стоит недооценивать важность данного шага. Соответствующие инструменты можно найти в разделе Reporting Tools (Средства подготовки отчетов) меню Applications (Приложения).

В большинстве случаев тесты на проникновение будут устроены совершенно по-разному, поскольку каждая организация может подвергаться разным угрозам и иметь различные ресурсы, которые требуется защищать. Kali Linux дает универсальную основу для решения подобных задач, именно здесь можно воспользоваться большим количеством функций по настройке Kali. Многие организации, которые выполняют такие исследования, поддерживают настроенные под свои нужды версии дистрибутива для внутреннего применения. Это позволяет им ускорить развертывание систем перед новым исследованием.

Среди часто встречающихся дополнительных настроек Kali Linux можно отметить следующие.

- Предустановка лицензированных коммерческих пакетов. Например, имеется пакет, такой как платный сканер уязвимостей, который планируется использовать в ходе многих сеансов тестирования на проникновение. Во избежание необходимости устанавливать этот пакет на каждой развернутой копии Kali можно интегрировать его в систему (https://docs.kali.org/kali-dojo/02-mastering-live-build).



Как результат, данный пакет окажется установленным при каждом развертывании Kali.

- Предварительно настроенная виртуальная частная сеть с обратным соединением (VPN). Это очень удобная функция для устройств, которые преднамеренно оставляют подключенными внутри исследуемой сети. Такие устройства позволяют проводить «удаленные внутренние» проверки. Устройство с функцией обратного соединения связывается с компьютером пентестера, создавая туннель, который можно использовать для подключения к внутренним системам. Дистрибутив Kali Linux ISO of Doom — пример как раз такой специальной настройки системы (https://www.offensivesecurity.com/kali-linux/kali-rolling-iso-of-doom/).
- Предустановленные инструменты и программы собственной разработки. Многие организации имеют наборы утилит собственной разработки, необходимые в ходе сеансов тестирования на проникновение, поэтому их предварительная установка при формировании специального образа системы позволяет экономить время.
- Предварительная настройка конфигурации ОС, в том числе отображения имен хостов на IP-адреса, обоев рабочего стола, настроек прокси-серверов и т. д. Многие пользователи Kali предпочитают особые настройки системы (https://www.offensive-security.com/kali-linux/kali-linux-recipes/). Если вы собираетесь регулярно переустанавливать систему, то сохранение подобных настроек может иметь смысл.

Оценка приложений

Большинство мероприятий по оценке защищенности систем отличаются достаточно большими масштабами. Особенностью же исследований приложений является тот факт, что изучению подвергается конкретная программа. Подобные проверки становятся все более распространенными из-за сложности жизненно важных приложений, используемых компаниями. Многие из таких приложений созданы собственными силами этих компаний. Если нужно, то исследование приложений может сопутствовать другим видам проверок. Среди видов приложений, которые могут быть проанализированы на предмет безопасности, можно отметить такие.

- Веб-приложения: эти приложения часто являются целями злоумышленников, поскольку, обычно обладая значительной поверхностью атаки, доступны из Интернета. Стандартные тесты нередко позволяют обнаружить базовые проблемы веб-приложений. Однако более детальное исследование, хотя и может занимать немало времени, позволяет найти скрытые дефекты программ. Для проведения подобных испытаний можно воспользоваться метапакетом `kali-linux-web`, который содержит большое количество полезных инструментов.
- Настольные приложения, распространяемые в виде исполняемых файлов: серверные приложения — не единственная цель нарушителей. Настольные приложения также подвержены атакам. В прошедшие годы многие настольные программы, такие как средства для чтения PDF-файлов или видеоприложения,

использующие интернет-ресурсы, подвергались множествам атак, что привело к их совершенствованию. Однако все еще имеется множество настольных приложений, в которых при правильном подходе можно найти массу уязвимостей.

- Мобильные приложения: с ростом популярности мобильных устройств эти приложения становятся постоянными предметами исследований безопасности. Такие приложения очень быстро развиваются и меняются, поэтому в данной сфере методология исследований пока не достигла достаточной зрелости, что ведет к регулярному, практически еженедельному, появлению новых методик. Инструменты, относящиеся к изучению мобильных приложений, можно найти в разделе меню приложений Kali Linux Reverse Engineering (Обратное проектирование).

Исследование приложений можно проводить самыми разными способами. Например, для идентификации потенциальных проблем подойдут автоматические средства, предназначенные для тестирования конкретного приложения. Основываясь на особенностях работы приложений, подобные средства пытаются найти в них неизвестные слабости, вместо того чтобы полагаться на набор заранее заданных сигнатур. Инструменты для анализа программ должны учитывать особенности их поведения. Вот, например, популярный сканер уязвимости веб-приложений Burp Suite (https://portswigger.net/burp/). В ходе исследования приложения он находит поля для ввода данных, после чего выполняет различные атаки методом SQL-инъекций, наблюдая в это время за приложением, чтобы выявить атаки, которые оказались успешными.

Существуют и более сложные сценарии анализа приложений. Такие проверки могут быть выполнены в интерактивном режиме. При их проведении используют модели *черного* и *белого ящиков*.

- Исследование методом черного ящика: инструмент (или исследователь) взаимодействует с приложением, не обладая специальными знаниями о нем или особым доступом к нему, превышающим возможности обычного пользователя. Например, в случае с веб-приложением исследователь может иметь только доступ к функциям и возможностям, открытым пользователю, не авторизованному в системе. Любая применяемая учетная запись будет такой же, которую рядовой пользователь может зарегистрировать самостоятельно. Это не позволит атакующему анализировать функционал, доступный только привилегированным пользователям, учетные записи которых необходимо создавать администратору.
- Исследование методом белого ящика: инструмент (или исследователь) часто имеет полный доступ к исходному коду приложения, административный доступ к платформе, на которой оно выполняется и т. д. Это гарантирует выполнение полного и тщательного анализа всех возможностей приложения независимо от того, где именно находится изучаемая функциональность. Минус такого исследования заключается в том, что оно не является имитацией реальных действий злоумышленника.

Конечно, между белым и черным есть и оттенки серого. Обычно то, как именно будет проходить работа с приложением, определяется целью исследования. Если

она заключается в определении того, что может произойти с приложением, которое окажется предметом целенаправленной внешней атаки, то, вероятно, лучше всего подойдет тестирование методом черного ящика. Если же цель состоит в идентификации и устранении как можно большего количества проблем с безопасностью за сравнительно короткое время, то исследование методом белого ящика способно оказаться более эффективным.

В других случаях можно применить гибридный подход, когда исследователь не обладает полным доступом к исходному коду приложения для платформы, на которой оно выполняется, но выданная ему учетная запись подготовлена администратором и открывает доступ к максимально возможному количеству функций приложения.

Kali — это идеальная платформа для всех подходов к исследованию приложений. После установки стандартного дистрибутива здесь можно найти множество сканеров, рассчитанных на конкретные приложения. Тут есть и инструменты для более продвинутых исследований. Среди них — редакторы исходного кода и сценарные окружения. В деле исследования приложений могут оказаться полезными материалы из разделов Web Application (Веб-приложения) и Reverse Engineering (Обратное проектирование) (https://tools.kali.org/category/reverse-engineering) сайта Kali Tools.

11.3. Формализация оценки

После подготовки окружения Kali и определения типа проверки вы почти готовы приняться за дело. Однако остался еще один шаг: формализация исследования. Он крайне важен, поскольку на этом шаге определяется то, чего именно ожидает от вас клиент. Кроме того, здесь вам выдают разрешение на проведение операций, которые в обычных условиях являются незаконными. Мы рассмотрим все это в общих чертах, но перед нами очень непростой и важный этап подготовки к исследованию систем на уязвимость, так что, вполне возможно, на данном этапе вам стоит посоветоваться с юристом вашей организации.

Частью процесса формализации является определение правил анализа, которых вы будете придерживаться в ходе работы. Эти правила касаются следующих моментов.

- С какими системами вы можете взаимодействовать? Важно иметь уверенность в том, что вы случайно не вмешаетесь в работу систем, жизненно важных для исследуемой компании.
- В какое время суток вы можете работать, с какой периодичностью можно проводить сеансы исследования системы? Некоторые организации предпочитают ограничивать число таких сеансов.
- При обнаружении потенциальной уязвимости можете ли вы ее эксплуатировать? Если нет, то каков процесс подтверждения наличия уязвимости? В некоторых организациях предпочитают жестко контролировать каждую попытку эксплуатации уязвимостей, в других используют подход, более или



менее реалистично имитирующий настоящую атаку. Лучше всего выяснить эти моменты заранее.

- Если обнаружена серьезная проблема, то как нужно поступить? Иногда организации ожидают немедленного оповещения о выявлении, в противном случае об этом обычно сообщают в конце исследования.
- С кем можно связаться в случае крайней необходимости или при возникновении каких-либо проблем? Это всегда важно знать.
- Каков перечень лиц, которые будут знать о том, что проводится исследование? Как информация об этом будет доведена до них? В ряде случаев организации хотят проверить реакцию их внутренних служб на происшествие, а также способность этих служб обнаруживать вторжения. Лучше всего знать об этом заранее, поскольку при таком подходе вы сможете при необходимости провести исследование скрытно.
- Чего ждет компания после завершения исследования? Как сообщить о результатах? Заранее выясните, что именно все заинтересованные лица ожидают от проверки. Четкое определение ожидаемых результатов — лучший способ избежать неоднозначных ситуаций после завершения работы.

Хотя данный список и не полон, он даст вам общее представление о тех вопросах, которые нужно решить до начала работы. Однако вам следует понимать, что без качественного оформления юридических формальностей тут не обойтись. После того как проверка формализована, необходимо получить соответствующие разрешения. Это важно, поскольку большинство действий, выполняемых в ходе анализа безопасности, могут оказаться нелегальными без разрешения соответствующего должностного лица компании.

После того как все вышеописанное согласовано, остается еще один важный шаг — проверка. Не доверяйте переданным вам материалам о границах исследования — всегда проверяйте их. Используйте несколько источников информации для подтверждения того, что системы, которые планируется проанализировать, находятся в собственности клиента и клиент ими управляет. Например, учитывая повсеместное применение облачных сервисов, организация попросту способна не учесть, что она не владеет предоставленными ей сервисами. Вы можете обнаружить, что вам требуется получить специальное разрешение от поставщика облачных услуг перед началом работы.

Кроме того, всегда проверяйте предоставленные вам блоки IP-адресов. Не полагайтесь на предположение организации о ее владении всем блоком, даже если вам сообщили о том, что для исследования подходит весь переданный вам диапазон адресов. Например, мы встречались с организациями, которые запрашивали анализ в диапазоне адресов целой сети класса C, в то время как им принадлежала лишь некая часть этого диапазона. Изучая всю сеть класса C, мы фактически атаковали бы соседей организации по адресному пространству. Подменю OSINT Analysis (OSINT-анализ) раздела Information Gathering (Сбор информации) содержит множество инструментов, которые могут вам помочь при проверке материалов для проведения исследований.

11.4. Типы атак

Итак, работа началась. Какие атаки проводятся в ходе проверки защищенности информационных систем? Каждому типу уязвимости (https://www.cvedetails.com/vulnerabilities-by-types.php) соответствует особый способ ее эксплуатации. В этом разделе мы расскажем о классах уязвимостей, с которыми вам придется сталкиваться чаще всего.

Неважно, какая именно категория уязвимостей вас интересует. Что бы это ни было, Kali упрощает поиск средств и эксплойтов. Меню Kali на рабочем столе разделено на категории, которые помогают найти подходящее средство. Кроме того, на сайте Kali Tools можно найти обширные перечни инструментов, доступных в Kali, организованных по категориям и для удобства снабженных тегами. Каждая страница средства содержит подробные сведения о нем, а также примеры его использования.

Атака типа «отказ в обслуживании» (DoS-атака)

Атаки на отказ в обслуживании (Denial of Service attack, DoS) используют уязвимости для блокировки работы сервисов, обычно приводя к остановке уязвимого процесса. Категория Stress Testing (Стресс-тестирование) в меню приложений Kali содержит множество инструментов, ориентированных на решение этой задачи.

Многие при встрече с термином «атака типа отказ в обслуживании» думают об атаках, потребляющих ресурсы, которые выполняются из множества источников, одновременно направленных на одну цель. Однако такая атака — это уже так называемая *распределенная* атака типа «отказ в обслуживании» (Distributed Denial Of Service Attack, DDoS). Подобные атаки редко являются частью профессионального исследования защищенности систем.

Вместо этого единичные DoS-атаки чаще всего выступают результатом неудачной попытки эксплуатации уязвимости. Если автор эксплойта выпустил частично функциональный код, доказывающий возможность атаки (Proof of Concept, PoC), и тот был использован кем-то на практике, то это может привести к ситуации, аналогичной DoS-атаке. Даже качественно написанный эксплойт способен работать только при совпадении множества специфических обстоятельств и приводить к отказу атакуемого сервиса в других случаях. Может показаться, что решением проблемы является применение только как следует проверенных эксплойтов или написание собственных эксплойтов. Однако, как бы там ни было, никаких гарантий при использовании эксплойтов нет, это ставит атакующего в жесткие рамки, приводя к неоправданным ограничениям, что ведет к смягчению анализа. Ключ к решению проблемы — компромисс. Не задействуйте PoC-эксплойты и непроверенный код при проведении реальных проверок и всегда следите за тем, чтобы юрист компании мог прикрыть вас от других неприятностей.

Обычно DoS-атаки не выполняют намеренно. Большинство средств автоматического обнаружения уязвимостей считают DoS-уязвимости низкорисковыми из-за того, что хотя атакующий может вывести из строя некий сервис, последний нельзя будет применять, например, для выполнения стороннего кода. Однако важно помнить о том, что не все эксплойты общедоступны и DoS-уязвимости

могут маскировать более глубокие и серьезные угрозы. Эксплойт, позволяющий выполнять произвольный код, пользуясь известной DoS-уязвимостью, может существовать, но не в публичном пространстве. Отсюда вытекает следующий вывод: обращайте внимание на DoS-уязвимости и рекомендуйте клиентам их патчить несмотря на то, что им, как правило, присваивают низкий уровень риска.

Нарушение целостности информации в памяти

Нарушение целостности информации в памяти происходит, когда некая область в памяти процесса случайно модифицируется из-за ошибки при разработке программы. Ошибки, связанные с памятью, обычно ведут к непредсказуемому поведению программ, однако во многих случаях позволяют манипулировать памятью процесса. Это дает атакующему возможность управлять потоком выполнения программы и совершать необходимые ему действия.

Подобные атаки обычно называют атаками типа «переполнение буфера», хотя это слишком упрощенный термин. Самые распространенные типы нарушения целостности информации в памяти сильно различаются, для их эксплуатации требуются особые подходы и технические приемы. Ниже представлены наиболее часто встречающиеся типы атак на память.

- Переполнение буфера в стеке: когда программа записывает в буфер, находящийся в стеке, больше информации, чем объем доступного пространства, данные в соседних участках памяти могут быть повреждены, что часто ведет к аварийному завершению работы программы.
- Повреждение памяти в куче: последняя выделяется во время выполнения программы и обычно содержит данные работающих процессов. Нарушение целостности данных в куче происходит из-за действий, направленных на перезапись памяти через указатели или связанные списки.
- Целочисленное переполнение памяти: этот вид переполнения возникает в том случае, когда приложение пытается создать числовое значение, которое нельзя поместить в выделенный для него участок памяти.
- Атака на функции форматирования строк: когда программа принимает, что ввел пользователь, и форматирует ввод без проверки информации, злоумышленник может узнать необходимые ему адреса памяти, либо данные в памяти могут быть перезаписаны. Это зависит от применяемых символов форматирования.

Атаки на веб-приложения

Из-за того что современные сайты — это уже давно не статичные документы, а динамически генерируемые для пользователя страницы, типичный сайт устроен очень сложно. Уязвимости веб-приложений коренятся в данной сложности. В ходе соответствующих атак целью служит либо серверная часть приложения, ответственная за создание страниц, либо сами страницы, которые видит посетитель сайта.

Веб-атаки чрезвычайно распространены, так как многие организации достигли уровня, на котором имеют очень мало общедоступных сервисов. Два наиболее часто встречающихся вида атак (https://www.owasp.org/index.php/Top_10_2013-Top_10) — SQL-инъекции и межсайтовый скриптинг (Cross-site Scripting, XSS).

- SQL-инъекции. Подобные атаки направлены на приложения, при разработке которых допущены ошибки в подсистемах проверки и очистки пользовательского ввода. Это ведет к возможности извлекать информацию из баз данных таких приложений или даже получать полный контроль над серверами.
- Межсайтовый скриптинг. Как и в случае с SQL-инъекциями, атаки, основанные на XSS, возможны из-за неправильного обращения с пользовательским вводом. Это позволяет взломщику манипулировать пользователем или сайтом, захватывая сессии и выполняя в браузере собственный код.

Часто веб-приложения бывают сложными, обладающими обширными возможностями, а порой и непростой для понимания логикой работы. Они представляют собой удобную мишень для злоумышленников. Раздел меню Web Application Analysis (Анализ веб-приложения) содержит полезные инструменты для проверки устойчивости веб-приложений к атакам. Кроме того, тут стоит обратить внимание на метапакет `kali-linux-web`.

Взлом паролей

Взлом паролей — это атаки на системы аутентификации различных сервисов. Такие атаки часто делят на онлайновые и офлайновые. В соответствии с этой классификацией устроен и раздел меню Password Attack (Взлом паролей). В ходе онлайновой атаки злоумышленник пытается войти в систему, перебирая множество паролей. При проведении офлайновой атаки проводится работа с хешированными или зашифрованными паролями, полученными взломщиком. Цель этой работы — раскрыть исходные пароли. Защита от офлайновых атак состоит в повышении сложности паролей, что увеличивает трудоемкость их раскрытия. Однако существуют методы, позволяющие подбирать даже очень сложные пароли, например, заключающиеся в использовании вычислений на мощных видеокартах, благодаря применению которых удается значительно повысить производительность программ-взломщиков. Метапакет `kali-linux-gpu` содержит множество инструментов, ориентированных на быстрый подбор паролей.

Чаще всего онлайн-атаки направлены на стандартные пароли, которые по умолчанию задают поставщики ПО. Поскольку эти пароли широко известны, то атакующий в надежде на успех проверит стандартные точки входа в приложения. Еще один вид таких атак, встречающийся довольно часто, — это атака по специально подготовленному словарю. В ходе ее создают список слов, учитывающий особенности целевого окружения, а затем выполняют онлайновую атаку, пытаясь, перебирая список, подобрать пароль к распространенным, стандартным или известным взломщику учетным записям.

В ходе тестирования очень важно понимать потенциальные последствия атак такого рода. Во-первых, обычно они очень заметны из-за повторяющихся попыток аутентификации. Во-вторых, такие атаки после множества попыток войти в некую учетную запись часто ведут к ее блокировке. И наконец, скорость выполнения подобных атак обычно невысока, что ведет к сложностям при необходимости перебирать словари паролей больших размеров.

Атаки на клиентские системы

Цель большинства атак — серверы, но ввиду того что серверные сервисы становится атаковать все сложнее, злоумышленники выбирают более легкие цели, например клиентские системы. При таком подходе атакующего интересуют различные приложения, установленные на компьютере сотрудника организации, которую он пытается взломать. Соответствующие инструменты, помогающие проводить такие атаки, можно найти в категории меню Social Engineering Tools (Инструменты социальной инженерии).

Подобные атаки эффективнее всего проводились в начале 2000-х, их целями были Flash, Adobe Reader и Java. В этих случаях злоумышленник попытается добиться посещения жертвой специально подготовленного сайта. Последний будет содержать особый код, который может воспользоваться уязвимостями в клиентских приложениях, что приведет к возможности запустить на целевой системе нечто, необходимое взломщику.

Атаки на клиентские системы невероятно сложно предотвращать. Тут многое зависит от обучения пользователей, постоянного обновления приложений и от сетевых средств контроля, позволяющих уменьшить риск.

11.5. Резюме

В этой главе мы кратко рассказали о роли Kali в области информационной безопасности. Мы поговорили о важности применения чистой установки системы, об использовании шифрования для того, чтобы в ходе реальных исследований обеспечить защиту данных клиента. Здесь же был поднят вопрос о важности грамотного юридического оформления проверок безопасности. Это позволяет защитить интересы пентестера и его клиента.

Компоненты модели CIA (confidentiality, integrity, availability — конфиденциальность, целостность, доступность) — это характеристики, на которые обращают особое внимание, занимаясь вопросами информационной безопасности. Мероприятия, направленные на соблюдение принципов CIA, являются частью стандартных процессов развертывания, поддержки и анализа систем. Понимание этих концепций поможет вам при идентификации жизненно важных компонентов систем и оценки объема сил и ресурсов, которые стоит вложить в исправление обнаруженных проблем.

Мы рассмотрели несколько типов уязвимостей информационных систем, таких как уязвимость к включению файлов, к SQL-инъекциям, к переполнению буфера, подверженность системы состоянию гонки.

Точность сигнатур уязвимостей крайне важна для того, чтобы получить максимальную отдачу от автоматического сканирования систем на уязвимости. Чем больше данных о системе удается собрать, тем выше шанс получить адекватные результаты такого сканирования. Именно поэтому очень популярно сканирование систем с предварительной аутентификацией.

Поскольку автоматические средства используют базы данных сигнатур для выявления уязвимостей, то любое, даже небольшое отклонение от известной сигнатуры может изменить результат и, соответственно, значимость обнаруженной уязвимости.

Мы разобрали четыре типа исследований информационной безопасности систем: *оценка уязвимости систем*, *оценка систем на соответствие стандартам безопасности*, *традиционное тестирование на проникновение* и *исследование приложений*. Для разных типов проверок характерен собственный набор инструментов, однако многие исследования используют одни и те же средства.

Оценка уязвимости систем сравнительно проста в сравнении с другими видами исследований. Часто она представляет собой автоматический сбор сведений о потенциальных уязвимостях целевого окружения. Мы выяснили, что уязвимость — это дефект информационной системы, с помощью которого можно нарушить ее конфиденциальность, целостность или доступность. Так как автоматический поиск уязвимостей основан на сигнатурах, этот тип проверки полагается на точность подобных сигнатур и способен давать ложноположительные и ложноотрицательные результаты. Основные инструменты для проведения подобных исследований можно найти в разделах Vulnerability Analysis (Анализ уязвимости) и Exploitation Tools (Средства эксплуатации уязвимостей) меню приложений Kali Linux.

Оценка систем на соответствие стандартам безопасности основана на индустриальных или государственных стандартах, которым должна соответствовать исследуемая организация. Среди таких стандартов можно отметить PCI DSS, DISA STIG и FISMA. Они, в свою очередь, строятся на основе наборов нормативных требований. Проверки систем на соответствие стандартам безопасности обычно начинаются с оценки уязвимостей.

Традиционное тестирование на проникновение — это тщательное исследование защищенности системы, которое предназначено для улучшения общего уровня безопасности организации и основано на изучении устойчивости систем к реальным угрозам. Подобные проверки включают несколько шагов (отраженных в структуре меню приложений Kali Linux) и завершаются попытками эксплуатации уязвимостей и получения доступа к наиболее защищенным компьютерам и сетям целевого окружения.

Исследование приложений, обычно проводимое в соответствии с моделями белого или черного ящика, направлено на конкретное приложение и предусматривает использование специализированных инструментов. Эти инструменты

можно найти в таких разделах меню приложений Kali, как Web Application Analysis (Анализ веб-приложения), Database Assessment (Исследование баз данных), Reverse Engineering (Обратное проектирование) и Exploitation Tools (Средства эксплуатации уязвимостей).

Мы рассмотрели несколько типов атак, устойчивость систем к которым проверяется при оценке их защищенности: атака типа «отказ в обслуживании», когда работа приложения нарушается и оно оказывается недоступным; атаки на память, направленные на манипуляцию памятью процессов, что часто позволяет атакующему запускать произвольный код; веб-атаки, направленные на веб-сервисы и выполняемые с помощью различных технологий наподобие SQL-инъекций и XSS; взлом паролей, в ходе которого часто используется техника подбора пароля к сервису по заранее сформированному списку.

ГЛАВА 12

Резюме: дальнейший путь



Ключевые темы:

- постоянные изменения;
- сертификаты;
- тренинги.

Поздравляем! Надеемся, теперь вы более близки с вашей системой Kali Linux и не побоитесь использовать ее для любого эксперимента, который придет вам в голову. Вы уже знакомы с ее самыми интересными функциями, но также знаете ее ограничения и различные способы их обойти.

Если вы не применяете все функции на практике, то сохраните эту книгу для справочных целей и освежите память, когда решитесь попробовать новую функцию. Помните, что для развития новых навыков нет ничего лучше, чем практика (и настойчивость). Старайтесь сильнее, как продолжают повторять тренеры Offensive Security (https://www.offensive-security.com/offsec/say-try-harder/).

12.1. Отслеживание изменений

С постоянно изменяющимся дистрибутивом, таким как kali-rolling, отдельные разделы руководства обязательно устареют. Мы сделаем все возможное, чтобы поддерживать его в актуальном состоянии (по крайней мере онлайн-версию), но для большинства разделов мы пытались предоставить общие объяснения, которые должны быть полезны в течение длительного времени.

Таким образом, вы будете готовы принять изменения и сможете найти решение любой появившейся проблемы. Лучше понимая Kali Linux и его связь с Debian, вы можете полагаться на сообщества Kali и Debian и их многочисленные ресурсы (баг-трекеры, форумы, рассылки и т. д.), если застрянете.

Не бойтесь регистрировать ошибки (см. раздел 6.3)! Если вы такой же, как я, то к моменту выполнения всех шагов, связанных с регистрацией хорошего отчета об ошибке (а это займет некоторое время), вы уже решите проблему или по крайней мере найдете неплохие наработки. И фактически зарегистрировав ошибку, вы поможете другим, кто столкнулся с данной проблемой.

12.2. Демонстрация новоприобретенных знаний

Вы гордитесь своими новыми навыками в Kali Linux? Хотели бы вы быть уверенными, что действительно помните важные вещи? Если ответите «да» на один из этих вопросов, то вам следует рассмотреть возможность подачи заявки на программу Certified Professional (сертифицированный профессионал) для Kali Linux.

Это всеобъемлющая сертификация, которая гарантирует, что вы знаете, как устанавливать и использовать Kali Linux во многих реалистичных вариантах применения. Это приятное дополнение к вашему резюме и также доказывает вашу готовность идти дальше.

12.3. Дальнейший путь

Эта книга научила вас многим вещам, которые должен знать любой пользователь Kali Linux, но нам пришлось принять несколько непростых решений, чтобы сократить ее; и кроме того, осталось много неохваченных тем.

Системное администрирование

Если вы хотите узнать больше о системном администрировании, то мы можем только порекомендовать вам ознакомиться с руководством администратора Debian (https://debian-handbook.info/get/).

Вы найдете там много дополнительных глав, охватывающих общие сервисы Unix, которые мы полностью пропустили в данной книге. В этом руководстве также даются дополнительные советы, в частности о системе пакетирования (которая также рассматривается более широко на самом низком уровне).

Книга Debian, очевидно, более глубоко отражает сообщество Debian и то, как оно организовано. Хотя эти знания не являются жизненно важными, они могут быть действительно полезны, когда вам приходится взаимодействовать с участниками Debian, например через отчеты об ошибках.

Тестирование на проникновение

Вы, наверное, уже заметили: эта книга не научила вас тестированию на проникновение. Но все, что вы узнали, тем не менее важно. Теперь вы готовы полностью использовать возможности Kali Linux, лучшего дистрибутива для пентестинга. И у вас есть основные навыки Linux, необходимые для участия в тренингах от Offensive Security.

Если вы чувствуете, что еще не готовы к оплачиваемому курсу, то можете начать с бесплатного онлайн-обучения Metasploit Unleashed. Это очень популярный инструмент пентестинга, и вы должны его знать при наличии у вас серьезных планов касательно изучения тестирования на проникновение.

Следующим логичным шагом было бы прохождение онлайн-курса «Тестирование на проникновение с Kali Linux» (https://www.offensive-security.com/information-security-training/), ведущего к знаменитой сертификации Offensive Security Certified Professional. Этот онлайн-курс можно выполнять в своем темпе, но сертификация на самом деле представляет собой сложный 24-часовой реальный практический тест на проникновение, который проходит в изолированной сети VPN.

Готовы ли вы принять вызов?

Об авторах

Разработчик Debian на протяжении более 20 лет и автор справочника администратора Debian **Рафаэль Херцог** — гуру Debian в команде Kali. Когда не работает с Kali, он готов поделиться своими экспертными знаниями по Debian через компанию Freexian, которую сам же основал. Он помогает другим людям, создавая деривативы, пользовательские установщики и программное обеспечение для пакетирования Debian, а также улучшая существующие пакеты (путем исправления ошибок и добавления новых функций) и т. д.

Мати Ахарони — специалист по информационной безопасности с более чем десятилетним активным участием в тематическом сообществе. Основал такие проекты, как BackTrack и Kali Linux с открытым исходным кодом, а также Exploit Database и Offensive Security — ведущую компанию в сфере информационной безопасности, известную своими отраслевыми сертификатами безопасности и курсами повышения квалификации. Между эксплуатацией и каталогизацией, тестированием на проникновение, развитием Kali и возней с оборудованием Ахарони увлеченно рассказывает всем, кто готов его слушать, о достоинствах Kali Linux.

Джим О'Горман — президент сервисов Offensive Security для США. Имеет более чем десятилетний опыт проведения тестов на проникновение в сильно защищенные среды по всему миру. Кроме того, является ведущим инструктором курса «Тестирование на проникновение с Kali Linux» компании Offensive Security.

Рафаэль Херцог, Джим О'Горман, Мати Ахарони

Kali Linux от разработчиков

Перевел с английского С. Черников

Заведующая редакцией	*Ю. Сергиенко*
Руководитель проекта	*О. Сивченко*
Ведущий редактор	*Н. Гринчик*
Литературный редактор	*Н. Хлебина*
Художественный редактор	*С. Заматевская*
Корректоры	*О. Андриевич, Т. Радецкая*
Верстка	*Г. Блинов*

Изготовлено в России. Изготовитель: ООО «Прогресс книга». Место нахождения и фактический адрес: 194044, Россия, г. Санкт-Петербург, Б. Сампсониевский пр., д. 29А, пом. 52. Тел.: +78127037373.

Дата изготовления: 08.2018. Наименование: книжная продукция. Срок годности: не ограничен.

Налоговая льгота — общероссийский классификатор продукции ОК 034-2014, 58.11.12 — Книги печатные профессиональные, технические и научные.

Импортер в Беларусь: ООО «ПИТЕР М», 220020, РБ, г. Минск, ул. Тимирязева, д. 121/3, к. 214, тел./факс: 208 80 01.

Подписано в печать 30.07.18. Формат 70×100/16. Бумага офсетная. Усл. п. л. 25,800. Тираж 1000. Заказ 0000.

Отпечатано в ОАО «Первая Образцовая типография». Филиал «Чеховский Печатный Двор».
142300, Московская область, г. Чехов, ул. Полиграфистов, 1.
Сайт: www.chpk.ru. E-mail: marketing@chpk.ru
Факс: 8(496) 726-54-10, телефон: (495) 988-63-87


ИЗДАТЕЛЬСКИЙ ДОМ
ПИТЕР®
WWW.PITER.COM